universität koblenz
Informatik

Cross-Domain Multimodal Knowledge Processing: Advanced Methods for Extraction, Classification, Prediction, and Interpretation

Habilitationsschrift

zur Erlangung der Habilitation als Privatdozent im Fachbereich Informatik

vorgelegt von
Dr. Zeyd Boukhers

Universität Koblenz - Fachbereich Informatik - Koblenz, Germany - 2026

Erklärung

Hiermit versichere ich, dass ich die vorliegende Habilitationsschrift selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Die Habilitationsschrift basiert auf einer Sammlung bereits veröffentlichter beziehungsweise zur Veröffentlichung eingereichter wissenschaftlicher Arbeiten. Soweit die Habilitationsschrift gemeinsam mit anderen Autorinnen und Autoren verfasste Arbeiten enthält, sind meine jeweiligen Eigenanteile in Abschnitt kenntlich gemacht.

Alle Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, wurden unter Angabe der Quellen kenntlich gemacht.

Generative KI-Werkzeuge wurden ausschließlich in begrenztem Umfang zur sprachlichen Überarbeitung, stilistischen Verbesserung sowie zur Korrektur von Grammatik und Formulierungen verwendet. Die wissenschaftlichen Inhalte, die Konzeption der Arbeiten, die methodischen Ansätze, die Durchführung der Analysen, die Interpretation der Ergebnisse sowie die inhaltliche Darstellung stammen vollständig von mir.

Ort, Datum Unterschrift

. . . . . .

Zusammenfassung

Diese Habilitationsschrift verfolgt ein einheitliches Forschungsprogramm im Bereich der domänenübergreifenden multimodalen Wissensverarbeitung: die Entwicklung und Evaluierung von Methoden zur Extraktion, Klassifikation, Vorhersage und Interpretation von Wissen aus textuellen und visuellen Daten in verschiedenen Anwendungsdomänen. Die zentrale These lautet, dass robuste Wissensverarbeitung Methoden erfordert, die *domänenadaptiv*, *multimodal bewusst* und *praktisch skalierbar* sind. Die Arbeit umfasst siebzehn Beiträge, die in verschiedenen Venues der Informationswissenschaft (JCDL, JASIST, TPDL), der medizinischen KI (NCA, BIBM, AIME), der Wissensverarbeitung und Sprachmodellforschung (CIKM, SIGDIAL, Entropy) sowie der erklärbaren KI (Sensors) veröffentlicht wurden und bis März 2026 über 190 Zitationen erzielt haben, einschließlich eines Best-Paper-Awards (TPDL 2022).

Die Beiträge sind in fünf Themen gegliedert. Im Bereich der *wissenschaftlichen Wissensinfrastruktur* verfolgt die Arbeit einen methodischen Bogen von klassischer CRF-basierter Sequenzetikettierung über tiefes Transfer-Lernen bis hin zu multimodalen räumlich-semantischen Encodern für die Metadaten-Extraktion aus wissenschaftlichen Dokumenten, ergänzt durch Beiträge zur Topic-Modellierung, zu datenschutzfreundlicher verteilter Analytik und zur LLM-basierten FAIR-Bewertung. Im Bereich der *Disambiguierung von Autorennamen* zeigt die Arbeit, dass bereits minimale bibliografische Metadaten, insbesondere Koautorschaften, Publikationstitel und Venue-Informationen, ausreichend Signal für eine effektive Disambiguierung in großskaligen Datenbanken wie DBLP mittels überwachter Lernverfahren bieten, wobei Herausforderungen bei neuen Koautorschaften und stark mehrdeutigen Namensvarianten bestehen bleiben.

Im Bereich der *domänenspezifischen Textklassifikation und -vorhersage* zeigt die Forschung, dass die Integration externer Wissensquellen entscheidender für die Leistung ist als die reine Modellgröße: wissensgestützte Architekturen und LLM-basierte Repräsentationen verbessern die ICD-Kodierung, und kleinere, domänenspezifisch feinabgestimmte LLMs können größere Allzweckmodelle in ressourcenbeschränkten klinischen Umgebungen erreichen oder übertreffen. Im Finanzbereich zeigt die Arbeit, dass öffentliche Aufmerksamkeit und Händlerstimmung signifikante Vorhersagekraft für Kryptowährungsvolatilität über traditionelle Handelsindikatoren hinaus bieten.

Im Bereich der *multimodalen Interpretierbarkeit* wird mit COIN eine kontrafaktische Bildgenerierungsmethode eingeführt, die entscheidende Bildregionen für VQA-Vorhersagen identifiziert, ohne Zugriff auf Modellinterna zu benötigen. Im Bereich der *Optimierung großer Sprachmodelle* löst das EMORL-Framework die Multi-Ziel-Feinabstimmung durch ensemble-basierte Aggregation verborgener Zustände, wodurch separate Trainingsläufe pro Ziel vermieden werden, und eine Studie zur Wissensdestillation zeigt, dass komprimierte Modelle über 90 % der Leistung des Lehrermodells beibehalten, während die Rechenanforderungen um bis zu 57 % reduziert werden.

Über die themenspezifischen Ergebnisse hinaus liefert die Arbeit drei übergreifende Erkenntnisse: Repräsentationsqualität ist durchweg entscheidender als architektonische Komplexität; die Integration heterogener Wissensquellen ist ein wiederkehrendes und domänenübergreifend erfolgreiches Muster; und der Übergang von LLMs als Werkzeuge zu LLMs als Untersuchungsgegenstand spiegelt eine produktive Forschungstrajektorie wider, die praktische Erfahrung mit methodischer Innovation verbindet. Gemeinsam demonstrieren diese Beiträge, dass domänenübergreifender Transfer methodischer Erkenntnisse substantielle Verbesserungen gegenüber domänenisolierten Ansätzen ermöglicht.

Abstract

This habilitation thesis pursues a unified research programme in cross-domain multimodal knowledge processing: the development and evaluation of methods that extract, classify, predict, and interpret knowledge from textual and visual data across multiple application domains. The central argument is that robust knowledge processing requires methods that are *adaptive to domain*, *aware of multiple modalities*, and *scalable in practice*. The work comprises seventeen contributions published in different venues across information science (JCDL, JASIST, TPDL), medical AI (NCA, BIBM, AIME), knowledge and language model research (CIKM, SIGDIAL, Entropy), and explainable AI (Sensors), accumulating over 190 citations by March 2026 and including a Best Paper Award (TPDL 2022).

The contributions are organised around five themes. In *scholarly knowledge infrastructure*, this work traces a methodological arc from classical CRF-based sequence labelling through deep transfer learning to multimodal spatial-semantic encoders for metadata extraction from scientific documents, complemented by contributions on topic modelling, privacy-preserving distributed analytics, and LLM-based FAIR assessment. In *author name disambiguation*, the work shows that even minimal bibliographic metadata, specifically co-authorships, publication titles, and venue information, carries sufficient signal for effective disambiguation in large-scale databases such as DBLP using supervised learning models, though challenges remain for cases involving new co-authorships or highly ambiguous name variants.

In *domain-specific text classification and prediction*, the research demonstrates that integration of external knowledge sources is more decisive for performance than model scale: knowledge-guided architectures and LLM-derived representations improve ICD coding, and smaller, domain-specifically fine-tuned LLMs can match or exceed larger general-purpose models in resource-constrained clinical settings. In finance, public awareness and trader sentiment provide significant predictive power for cryptocurrency volatility beyond traditional trading indicators. In *multimodal interpretability*, this work introduces COIN, a counterfactual image generation method that identifies decisive image regions for VQA predictions without requiring access to model internals. In *large language model optimisation*, the EMORL framework addresses multi-objective fine-tuning through ensemble-based hidden-state aggregation, avoiding separate training runs per objective, and a knowledge distillation study shows that compressed models retain over 90% of teacher performance while reducing computational requirements by up to 57%.

Beyond the theme-specific findings, the work yields three programme-level insights: representation quality is consistently more decisive than architectural complexity; the integration of heterogeneous knowledge sources is

a recurring and cross-domain successful pattern; and the progression from using LLMs as tools to studying them as objects reflects a productive research trajectory that connects practical experience with methodological innovation. Together, these contributions demonstrate that cross-domain transfer of methodological insights yields substantial improvements over domain-isolated approaches.

Publications and Disclaimer

This thesis is based on seventeen (17) contributions published in different venues. Throughout this document, these contributions are referred to by the labels P1 through P17, corresponding to their order in the list below. Table 1 provides a consolidated overview of all contributions with their venues, venue quality indicators, citation counts, and contribution types; detailed scope and own-contribution statements follow in the full list.

Table 1: Overview of the seventeen contributions (P1–P17) grouped by thesis theme. Venue ranks refer to CORE 2023 for conferences and SJR/JCR for journals. Citation counts are from Google Scholar (March 2026).

ID	Short title	Venue	Rank	Cit.	Role
Theme 1: Scholarly Knowledge Infrastructure					
P1	High-variance reference extraction	JCDL 2019	A*	23	Lead
P2	MexPub (German metadata extraction)	JCDL 2021	A*	17	Lead
P3	Multimodal metadata extraction	JCDL 2022	A*	10	Lead
P4	TextMap (spatial-semantic framework)	JASIST (u.r.)	Q1	0	Lead
P14	Rényi entropy for topic modelling	Entropy 2020	Q1	20	Collab.
P15	EXCITE toolchain	JCDL 2019	A*	23	Collab.
P16	PADME-SoSci	JCDL 2023	A*	3	Lead
P17	FAIR-Way (LLM-based FAIR assessment)	CIKM 2025	A	0	Collab.
Theme 2: Author Name Disambiguation					
P5	Whois? Deep disambiguation [†]	TPDL 2022	B	15	Lead
P6	DBLP-based disambiguation	IJDL 2023	Q2	14	Lead
Theme 3: Domain-Specific Classification and Prediction					
P7	Knowledge-guided ICD coding	NCA 2023	Q1	7	Lead
P8	LLMs for ICD coding	BIBM 2024	–	6	Lead
P9	Cryptocurrency volatility and sentiment	CIKM 2023	A	22	Lead
P13	ELMTEX (clinical extraction with LLMs)	AIME 2025	B	9	Collab.
Theme 4: Multimodal Interpretability					
P10	COIN (counterfactual VQA)	Sensors 2022	Q1	15	Lead
Theme 5: LLM Optimisation					
P11	EMORL (ensemble multi-objective RL)	SIGDIAL 2025	B	6	Superv.
P12	Knowledge distillation for QA	SIGDIAL 2025	B	0	Superv.

[†]Best Paper Award, TPDL 2022. *Role*: Lead = first and corresponding author; Superv. = supervised contribution (PhD/Master's student); Collab. = collaborative contribution. *u.r.* = under review.

The seventeen contributions are listed in detail below.

1. **Zeyd Boukhers**(✉), Shriharsh Ambhore, and Steffen Staab. "*An End-to-End Approach for Extracting and Segmenting High-Variance References from PDF Documents.*" In: *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2019. [46]

    - **Conference CORE ranking:** A*
    - **Paper citation count:** 23 (March 2026)
    - **Paper type:** Long conference paper
    - **Scope:** This paper presents an end-to-end approach for extracting and segmenting highly variable references from PDF documents, addressing noisy layouts and heterogeneous reference structures in scholarly collections.
    - **Own contribution:** I conceived the study, led the methodological development and implementation, carried out the experimental evaluation, and wrote the manuscript.

2. **Zeyd Boukhers**(✉), Nabil Beili, Timo Hartmann, Prantik Goswami, and Muhammad Awais Zafar. "*MexPub: Deep Transfer Learning for Metadata Extraction from German Publications.*" In: *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2021. [49]

    - **Conference CORE ranking:** A*
    - **Paper citation count:** 17 (March 2026)
    - **Paper type:** Short conference paper
    - **Scope:** This paper introduces a deep transfer learning approach for metadata extraction from German scholarly publications and shows how transfer learning improves extraction quality in comparatively low-resource settings.
    - **Own contribution:** I initiated the work, developed the main methodological approach, supervised the implementation and evaluation, and led the writing of the manuscript.

3. **Zeyd Boukhers**(✉), and Azeddine Bouabdallah. "*Vision and Natural Language for Metadata Extraction from Scientific PDF Documents: A Multimodal Approach.*" In: *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*. 2022. [52]

    - **Conference CORE ranking:** A*
    - **Paper citation count:** 10 (March 2026)
    - **Paper type:** Short conference paper

- **Scope:** This paper proposes a multimodal approach for metadata extraction from scientific PDF documents by combining visual and textual representations and demonstrates the benefit of jointly modelling document layout and language.
- **Own contribution:** I formulated the research question, designed the multimodal framework, conducted the main experimental analysis, and wrote the paper.

4. **Zeyd Boukhers**(✉), and Cong Yang. "*TextMap: A Spatial-Semantic Framework for PDF Metadata Extraction and Comparative Performance Analysis.*" *Journal of the Association for Information Science and Technology*, 2025. [57]

   - **Journal quartile ranking:** Q1
   - **Journal impact factor:** 4.3
   - **Paper citation count:** 0 (March 2026)
   - **Paper type:** Regular research article
   - **Status:** Under review
   - **Scope:** This paper introduces a spatial-semantic framework for PDF metadata extraction and provides a comparative analysis of different feature learning strategies for robust metadata recognition across diverse document layouts.
   - **Own contribution:** I conceived the study, developed the framework, carried out the comparative evaluation, and led the manuscript preparation.

5. **Zeyd Boukhers**(✉), and Nagaraj Bahubali Asundi. "*Whois? Deep Author Name Disambiguation Using Bibliographic Data.*" In: *International Conference on Theory and Practice of Digital Libraries*. Cham: Springer International Publishing, 2022. [47]

   - **Conference CORE ranking:** B
   - **Paper citation count:** 15 (March 2026)
   - **Paper type:** Long conference paper
   - **Scope:** This paper presents a deep learning approach for author name disambiguation based on bibliographic metadata and demonstrates how metadata can support author identity resolution in digital library settings.
   - **Own contribution:** I defined the research problem, designed the modelling approach, supervised the experiments, and led the writing of the paper.

6. **Zeyd Boukhers**(✉), and Nagaraj Bahubali Asundi. "*Deep Author Name Disambiguation Using DBLP Data.*" *International Journal on Digital Libraries* (2023): 1–11. [48]

    - **Journal quartile ranking:** Q2
    - **Journal impact factor:** 1.6
    - **Paper citation count:** 14 (March 2026)
    - **Paper type:** Regular research article
    - **Scope:** This paper extends the author disambiguation line of research by studying deep learning for author identity resolution on DBLP data and by analysing the utility of bibliographic metadata for this task.
    - **Own contribution:** I led the study design, methodological development, evaluation, and manuscript writing.

7. **Zeyd Boukhers**(✉), Prantik Goswami, and Jan Jürjens. "*Knowledge Guided Multi-Filter Residual Convolutional Neural Network for ICD Coding from Clinical Text.*" *Neural Computing and Applications* 35.24 (2023): 17633–17644. [54]

    - **Journal quartile ranking:** Q1
    - **Journal impact factor:** 4.5
    - **Paper citation count:** 7 (March 2026)
    - **Paper type:** Regular research article
    - **Scope:** This paper proposes a knowledge-guided residual convolutional neural architecture for automatic ICD coding from clinical text, combining textual representations with external knowledge to improve classification performance.
    - **Own contribution:** I conceived the methodological idea, led the modelling and evaluation, and wrote the manuscript.

8. **Zeyd Boukhers**(✉), AmeerAli Khan, Qusai Ramadan, and Cong Yang. "*Large Language Model in Medical Informatics: Direct Classification and Enhanced Text Representations for Automatic ICD Coding.*" In: *International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2024. [56]

    - **Conference venue:** IEEE International Conference on Bioinformatics and Biomedicine (BIBM)
    - **Paper citation count:** 6 (March 2026)
    - **Paper type:** Short conference paper

- **Scope:** This paper investigates the use of large language models for automatic ICD coding, studying both direct classification and the use of LLM-derived text representations in downstream architectures.
- **Own contribution:** I initiated the work, defined the study design, guided the experiments, and led the writing.

9. **Zeyd Boukhers**(✉), Azeddine Bouabdallah, Cong Yang, and Jan Jürjens. "*Beyond Trading Data: The Hidden Influence of Public Awareness and Interest on Cryptocurrency Volatility.*" In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2023. [53]

    - **Conference CORE ranking:** A
    - **Paper citation count:** 22 (March 2026)
    - **Paper type:** Long conference paper
    - **Scope:** This paper studies how public awareness and user interest, reflected in textual and behavioural signals, affect cryptocurrency volatility, extending text and sentiment analysis to financial forecasting.
    - **Own contribution:** I conceived the study, designed the methodological setup, conducted the main analyses, and led the paper writing.

10. **Zeyd Boukhers**(✉), Timo Hartmann, and Jan Jürjens. "*COIN: Counterfactual Image Generation for Visual Question Answering Interpretation.*" *Sensors* 22.6 (2022): 2245. [55]

    - **Journal quartile ranking:** Q1
    - **Journal impact factor:** 3.4
    - **Paper citation count:** 15 (March 2026)
    - **Paper type:** Regular research article
    - **Scope:** This paper introduces a counterfactual image generation approach for interpreting visual question answering models, identifying image regions that are decisive for model predictions.
    - **Own contribution:** I initiated the work, developed the conceptual approach, supervised the implementation and evaluation, and wrote the manuscript.

11. Lingxiao Kong, Cong Yang, Susanne Neufang, Oya Deniz Beyan, **Zeyd Boukhers**(✉). "*EMORL: Ensemble Multi-Objective Reinforcement Learning for Efficient and Flexible LLM Fine-Tuning.*" In: *Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 2025. [192]

- **Conference CORE ranking:** B
- **Paper citation count:** 6 (March 2026)
- **Paper type:** Long conference paper
- **Scope:** This paper proposes an ensemble multi-objective reinforcement learning framework for efficient and flexible fine-tuning of large language models under multiple optimisation criteria.
- **Own contribution:** I contributed to the conceptual framing of the work, the experimental design, the interpretation of the results, and the preparation of the manuscript.

12. Joyeeta Datta, Niclas Doll, Qusai Ramadan, **Zeyd Boukhers**(✉). "*Exploring the Limits of Model Compression in LLMs: A Knowledge Distillation Study on QA Tasks.*" In: *Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 2025. [87]

    - **Conference CORE ranking:** B
    - **Paper citation count:** 0 (March 2026)
    - **Paper type:** Short conference paper
    - **Scope:** This paper studies knowledge distillation for compressing large language models on question answering tasks and analyses the trade-off between compression and downstream performance.
    - **Own contribution:** I contributed to the study design, methodological discussion, evaluation strategy, and manuscript revision.

13. Aynur Guluzade, Naguib Heiba, **Zeyd Boukhers**, Florim Hamiti, Jahid Hasan Polash, Yehya Mohamad, and Carlos A. Velasco. "*ELM-TEX: Fine-Tuning LLMs for Structured Clinical Information Extraction. A Case Study on Clinical Reports.*" In: *International Conference on Artificial Intelligence in Medicine*. 2025. [137]

    - **Conference CORE ranking:** B
    - **Paper citation count:** 9 (March 2026)
    - **Paper type:** Short conference paper
    - **Scope:** This paper studies fine-tuning large language models for structured clinical information extraction from clinical reports, benchmarking models of different sizes and showing that smaller fine-tuned models can match or exceed larger models in resource-constrained settings.
    - **Own contribution:** I contributed to the conceptual framing of the clinical information extraction problem, the methodological design and evaluation strategy, and the interpretation and revision of the manuscript.

14. Sergei Koltcov, Vera Ignatenko, **Zeyd Boukhers**, and Steffen Staab. "*Analyzing the Influence of Hyper-Parameters and Regularizers of Topic Modeling in Terms of Rényi Entropy.*" *Entropy* (2022): [191]

    - **Journal quartile ranking:** Q1
    - **Paper citation count:** 20 (March 2026)
    - **Paper type:** Regular research article
    - **Scope:** This paper analyses how hyper-parameters and regularisation choices influence topic modelling behaviour using Rényi entropy, contributing to the understanding of semantic structure learning in text corpora.
    - **Own contribution:** I contributed to the methodological analysis, the interpretation of the results, and the preparation of the manuscript.

15. Azam Hosseini, Behnam Ghavimi, **Zeyd Boukhers**, and Philipp Mayr. "*EXCITE–A Toolchain to Extract, Match and Publish Open Literature References.*" In: *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2019. [158]

    - **Conference CORE ranking:** A*
    - **Paper citation count:** 23 (March 2026)
    - **Paper type:** Demo paper
    - **Scope:** This paper presents the EXCITE toolchain for extracting, matching, and publishing open literature references, supporting scalable scholarly reference processing and reuse.
    - **Own contribution:** I contributed to the system conception, the technical implementation, and the presentation of the demonstrator.

16. **Zeyd Boukhers**, Arnim Bleier, Yeliz Ucer Yediel, Mio Hienstorfer-Heitmann, Mehrshad Jaberansary, Adamantios Koumpis, and Oya Beyan. "*PADME-SoSci: A Platform for Analytics and Distributed Machine Learning for the Social Sciences.*" In: *2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2023. [51]

    - **Conference CORE ranking:** A*
    - **Paper citation count:** 3 (March 2026)
    - **Paper type:** Demo paper
    - **Scope:** This paper presents PADME-SoSci, a platform for privacy-preserving distributed analytics and machine learning in the social sciences, enabling cross-site model training while maintaining data locality and ownership.

- **Own contribution:** I initiated and coordinated the work, contributed to the platform design, and led the preparation of the demonstrator paper.

17. Anmol Sharma, Sulayman K. Sowe, Soo-Yon Kim, Sayed Hoseini, Fidan Limani, **Zeyd Boukhers**, Christoph Lange, and Stefan Decker. "*FAIR Data Assessment Using LLMs: The Fair-Way.*" In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2025. [314]

    - **Conference CORE ranking:** A
    - **Paper citation count:** 0 (March 2026)
    - **Paper type:** Conference paper
    - **Scope:** This paper introduces an LLM-based approach for automated FAIRness assessment of metadata, using decomposition into fine-grained tasks to improve interpretability and generalisability across FAIR assessment settings.
    - **Own contribution:** I contributed to the conceptual framing, the methodological discussion, the evaluation design, and the refinement of the manuscript.

where ✉ denotes the corresponding and leading author.

Author Contributions and Scope of the Main Body

Primary contributions (P1–P10). I am the primary contributor to Contributions P1 through P10, having carried out at least 80% of the research, methodological development, implementation, experimental evaluation, and manuscript preparation for each of these works. In all of these contributions, I served as both the first author and the corresponding author, and I led the overall research direction.

Supervised contributions (P11 and P12). Contributions P11 and P12 originate from research conducted under my direct supervision. P11 is primarily the work of my PhD student, whom I supervise on a daily basis, while P12 is based on research carried out by my Master's student under my close co-supervision. In both cases, I contributed approximately 50% to the conceptual development, methodological design, evaluation strategy, and scientific writing.

Collaborative contributions (P13–P17). Contributions P13 through P17 represent additional collaborative research outputs in which I played a significant intellectual, methodological, and organisational role, including problem formulation, study design, implementation, supervision, acquisition of resources, interpretation of results, and manuscript preparation.

Scope of the main body. To avoid redundancy and to maintain a coherent scientific narrative, the main body of this thesis reproduces in full *only the contributions that represent the current state of each research line.* Specifically:

- P4 subsumes and substantially extends P2 and P3 in the metadata extraction line of Theme 1. Accordingly, P4 is reproduced in full, while P2 and P3 are not reproduced separately. Their specific methodological contributions are summarised and referenced within the presentation of P4 and in Section 2.2.
- P6 is a substantially extended journal version of P5, which received the Best Paper Award at TPDL 2022. P6 is reproduced in full, while P5 is not reproduced separately. The results and methodological steps introduced in P5 are referenced where relevant.

All other contributions (P1, P4, P6–P17) are reproduced in full in the subsequent sections of this thesis. The contributions that are not reproduced in full (P2, P3, P5) remain part of the scientific evidence supporting this habilitation and are cited throughout the framework.

Research Impact Overview

- **Publication Quality:**
  - **Digital Libraries and Information Science:** 4 A* JCDL papers, 1 TPDL paper (CORE B), 1 *JASIST* paper (Q1, under review), and 1 *IJDL* paper (Q2)
  - **Medical and Clinical AI:** 1 *Neural Computing & Applications* paper (Q1), 1 BIBM paper, and 1 AIME paper (CORE B)
  - **AI, Language, and Knowledge Management:** 2 SIGDIAL papers (CORE B), 2 CIKM papers (CORE A), and 1 *Entropy* paper (Q1)
  - **Visual and Explainable AI:** 1 *Sensors* paper (Q1)
- **Research Impact:**
  - Total citations across the 17 contributions: 190+ (Google Scholar, March 2026)
  - Best Paper Award (TPDL 2022, Contribution P5)

Acknowledgments

I am deeply grateful to my collaborators at the "*Institute of Web and Data Science*" of the University of Koblenz and the "*FAIR Data and Distributed Analytics*" group at Fraunhofer FIT.

I owe a particular debt of gratitude to *Prof. Steffen Staab* for his long-standing scientific mentorship and to *Prof. Jan Jürjens* for our fruitful collaborations over the years, both of which are reflected in several contributions of this thesis. I also thank *Prof. Matthias Thimm*, *Prof. Frank Hopfgartner*, and *Dr. Christoph Lange* for their support as heads of the institutes/departments in which this work took shape. I am especially grateful to *Prof. Jürjens* and *Prof. Hopfgartner* for additionally serving as reviewers of this thesis.

A substantial part of this thesis grew out of research carried out together with the students I have had the privilege to supervise. I warmly thank *Anmol Sharma*, *Nagaraj Bahubali Asundi*, *Azeddine Bouabdallah*, *Joyeeta Datta*, *Prantik Goswami*, *Timo Hartmann*, *AmeerAli Khan*, and *Lingxiao Kong* for their dedication, curiosity, and the excellent work that became part of these contributions.

I further thank all co-authors of the contributions collected in this thesis, listed in the Publications section, whose expertise and collaboration enriched this work immensely.

My deepest gratitude goes to my wife, whose unwavering support and understanding made this journey possible, and to our children, who bring joy and purpose to everything I do. I am forever indebted to my parents for their constant encouragement and support throughout my academic journey. I also thank my brothers and my sister for always being there.

This thesis is dedicated to the loving memory of my grandfather *Ahmad*, and my aunts *Fafa* and *Djazia*, whose passing in 2023/2024 left an immeasurable void.

Contents

1 Introduction

The extraction of actionable knowledge from documents, whether scholarly articles, clinical reports, or financial communications, remains one of the central challenges in information science and computational knowledge management. Despite decades of progress in natural language processing (NLP) and machine learning (ML), several obstacles persist. Documents exhibit highly heterogeneous layouts, making robust information extraction difficult. Domain-specific terminology introduces ambiguity, complicating tasks from medical coding to author identity resolution. Relevant signals are often distributed across modalities such as text, images, and document layout, requiring models that can fuse these representations effectively. The growing complexity of the resulting models raises questions of interpretability: understanding *why* a model produces a given prediction is essential for trust and accountability. Finally, the models themselves, particularly large language models (LLMs), are becoming so resource-intensive that their efficient training and deployment have emerged as research problems in their own right. This habilitation thesis addresses these challenges through a unified research programme centred on **cross-domain multimodal knowledge processing**: the development and evaluation of methods that extract, classify, predict, and interpret knowledge from textual and visual data across multiple application domains.

The central argument of this work is that robust knowledge processing requires methods that are *adaptive to domain*, *aware of multiple modalities*, and *scalable in practice*. No single technique suffices: extracting metadata from a scanned German social science article demands different representations than predicting cryptocurrency volatility from trader tweets, yet both tasks share underlying challenges of noisy input, limited labelled data, and the need to integrate heterogeneous information sources. By systematically investigating these shared challenges across six application areas, namely information studies, healthcare, finance, explainable AI, and large language model optimisation, this thesis demonstrates that cross-domain transfer of methodological insights yields substantial improvements over domain-isolated approaches.

The seventeen contributions presented in this thesis are organised along five dimensions, as illustrated in the taxonomy in Figure 1: *domain*, *task*, *data type*, *approach*, and *modality*. This taxonomy is not merely a cataloguing device; it reflects a deliberate research design in which each contribution addresses a specific combination of domain and task while building on methodological advances from the others. The following paragraphs introduce the five research themes that structure this thesis, motivate the key challenges in each, and preview the contributions that address them.

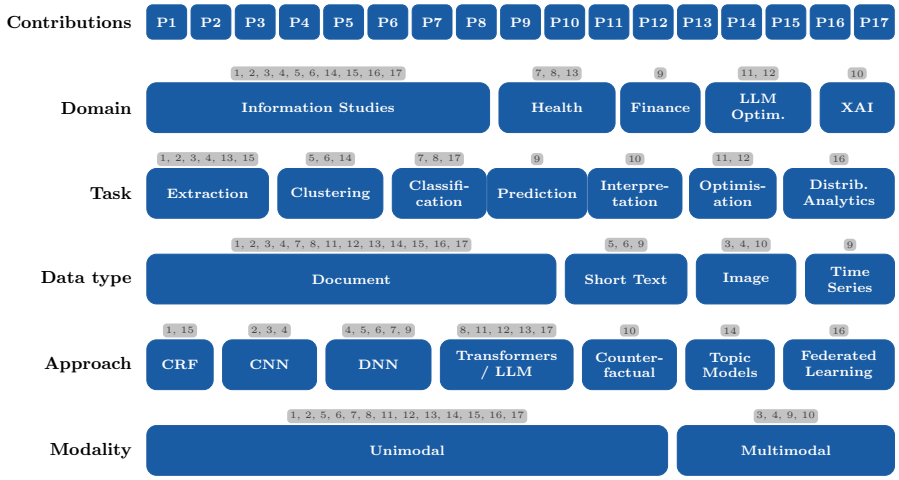


Figure 1: Taxonomy of contributions, domains, tasks, data types, approaches, and modalities. The numbers associated with each box refer to the specific contributions (P1 [46], P2 [49], P3 [52], P4 [57], P5 [47], P6 [48], P7 [54], P8 [56], P9 [53], P10 [55], P11 [192], P12 [87], P13 [137], P14 [191], P15 [158], P16 [51], P17 [314]).

Theme 1: Scholarly Knowledge Infrastructure. The FAIR principles [361] have established that scholarly data must be findable, accessible, interoperable, and reusable, yet a prerequisite for FAIRness is the availability of structured metadata, which remains absent for a large fraction of the scholarly record. In the EXCITE project[1], for instance, approximately 56% of articles in German social sciences published before 2016 lacked structured metadata. European and German initiatives such as EOSC[2] and NFDI[3] are investing heavily in research data infrastructures to close this gap, but scalable automated methods are essential to processing the existing backlog.

This thesis addresses the scholarly knowledge infrastructure challenge at three levels: *extraction*, *analysis*, and *assessment*. At the extraction level, four contributions trace a methodological arc from classical sequence labelling to multimodal deep learning. Contribution P1 [46] introduces an end-to-end CRF-based pipeline for extracting and segmenting highly variable references from PDF documents. Contribution P2 [49] advances this with deep transfer learning, demonstrating improved extraction from German publications in low-resource settings. Contribution P3 [52] adds a visual modality, combining document layout features with textual representations in a multimodal framework. Contribution P4 [57] presents a spatial-semantic

[1] `https://excite.informatik.uni-stuttgart.de/`
[2] `https://eosc.eu/`
[3] `https://www.nfdi.de/`

framework and a systematic comparative analysis of feature learning strategies, consolidating the insights of the preceding work. These extraction methods are operationalised in the EXCITE toolchain (P15 [158]), which provides an end-to-end system for extracting, matching, and publishing open literature references at scale.

At the analysis level, Contribution P14 [191] investigates how hyperparameters and regularisation choices influence topic modelling behaviour, analysed through the lens of Rényi entropy. This work contributes to the understanding of semantic structure learning in text corpora, a foundation that informs the representation choices made throughout this thesis.

However, applying analytical methods to scholarly data is not only a methodological challenge but also an institutional one: research datasets are frequently distributed across organisations that cannot share raw data due to privacy or governance constraints. Contribution P16 [51] addresses this barrier by presenting PADME-SoSci, a platform for privacy-preserving distributed machine learning in the social sciences. By enabling cross-site model training while maintaining data locality, PADME-SoSci provides the infrastructure layer that allows the extraction and analysis methods developed in this thesis to operate across institutional boundaries.

At the assessment level, Contribution P17 [314] closes the loop by introducing an LLM-based approach for automated FAIRness assessment of metadata, decomposing the evaluation into fine-grained tasks to improve interpretability and generalisability. Together, these contributions form a coherent pipeline from metadata extraction through analytical infrastructure to quality assessment, addressing the scholarly knowledge lifecycle end to end.

Theme 2: Author Name Disambiguation. Accurate metadata is a necessary but not sufficient condition for reliable scholarly analytics; the identity of the authors behind each record must also be resolved. Author name disambiguation is complicated by the prevalence of shared names and the limited distinguishing information available in bibliographic records. Contributions P5 [47] and P6 [48] address this challenge by training deep neural network models on minimal metadata, specifically article titles and co-authorship graphs, to disambiguate authors in large-scale databases such as DBLP. The key insight is that even short metadata strings carry sufficient signal for high-accuracy disambiguation when processed with appropriate representation learning, directly extending the metadata extraction and semantic representation advances of Theme 1.

Theme 3: Domain-Specific Text Classification and Prediction. A recurring challenge across application domains is that domain texts are long, terminologically dense, and associated with large, imbalanced label spaces.

This thesis addresses three instances of this challenge across healthcare and finance.

In healthcare, automatic ICD coding requires mapping clinical discharge summaries to a vast and fine-grained code taxonomy, a task complicated by code sparsity and overlapping medical terminology. Contribution P7 [54] proposes a knowledge-guided residual convolutional architecture that enriches text representations with external medical knowledge to mitigate data sparsity. Contribution P8 [56] extends this line by investigating LLMs (specifically LLaMA) as both direct classifiers and text representation generators for downstream ICD coding architectures, showing that LLM-derived representations substantially improve classification precision. Complementing these contributions, Contribution P13 [137] studies fine-tuning of LLMs for structured clinical information extraction from medical reports, demonstrating that smaller, task-specifically fine-tuned models can match or exceed larger general-purpose models in resource-constrained clinical settings. Taken together, P7, P8, and P13 illustrate a progression from knowledge-augmented convolutional models to LLM-based architectures for clinical NLP, with a consistent finding that *domain adaptation and external knowledge integration* are more decisive than model scale alone.

In finance, Contribution P9 [53] analyses the influence of public awareness and trader sentiment, captured from tweets and behavioural signals, on cryptocurrency market volatility. This work demonstrates that textual sentiment features, when integrated with trading data and temporal indicators, provide significant predictive power beyond traditional market variables alone.

Across both domains, the common methodological thread is the *integration of heterogeneous knowledge sources*, including external medical ontologies, LLM-generated representations, and social media signals, to compensate for the limitations of the primary textual data.

Theme 4: Multimodal Interpretability. As the models developed in the preceding themes grow in complexity, understanding *why* a model produces a given output becomes increasingly important. Contribution P10 [55] addresses this in the domain of Visual Question Answering (VQA) by introducing a counterfactual image generation method (COIN) that identifies which image regions are decisive for a model's prediction. By producing minimal image modifications that change the predicted answer, COIN provides interpretable explanations without requiring access to model internals, thereby contributing to the transparency and trustworthiness of multimodal AI systems.

Theme 5: Large Language Model Optimisation. Several of the preceding contributions employ LLMs as components of their pipelines, most

directly P8 and P13 in healthcare, and P17 for FAIR assessment. However, the computational cost of training and deploying these models poses practical barriers. This thesis addresses two complementary aspects of LLM optimisation. Contribution P11 [192] introduces EMORL, an ensemble multi-objective reinforcement learning framework that balances competing objectives such as fluency, factual accuracy, and domain relevance during fine-tuning, achieving efficient Pareto-optimal training without requiring separate runs per objective. Contribution P12 [87] investigates knowledge distillation for model compression on question-answering tasks, showing that student models retain over 90% of teacher performance while reducing computational requirements by up to 57%. Together, these contributions address the deployment gap that separates LLM research from real-world application, a gap that is directly felt in the clinical and scholarly settings of Themes 1 through 3.

Research Questions and Structure. The five themes above are guided by the following overarching research questions:

RQ1: How can scholarly knowledge infrastructure be strengthened through automated metadata extraction, semantic analysis of text corpora, privacy-preserving distributed analytics, and LLM-based quality assessment?

RQ2: To what extent can minimal bibliographic metadata support accurate author name disambiguation in large-scale scholarly databases?

RQ3: How can external knowledge sources and large language model representations be leveraged to improve text classification in domains characterised by large label spaces and data sparsity, and does domain-specific fine-tuning outperform sheer model scale?

RQ4: How can counterfactual explanations enhance the interpretability of multimodal deep learning models?

RQ5: What are effective strategies for optimising the training and compressing the deployment of large language models without significant performance loss?

The remainder of this thesis is structured as follows. Section 2 provides the theoretical background and positions each theme within the relevant literature. The full text of the primary contributions follows in the subsequent sections 3-16. Section 17 synthesises the findings and discusses future research directions.

2 Background and Theoretical Framework

This section provides the theoretical background for the five research themes introduced in Section 1. Section 2.1 presents the shared computational foundations that underpin all contributions, covering text representation, the transformer architecture, and the pre-train / fine-tune paradigm. Sections 2.2 through 2.6 then address each theme in turn, reviewing the relevant state of the art, identifying the gaps that motivate the contributions, and positioning each contribution within the existing literature.

2.1 Shared Computational Foundations

The contributions in this thesis span multiple domains, yet they share a common set of computational building blocks. This subsection introduces these foundations once to avoid repetition in the theme-specific sections that follow.

2.1.1 From Discrete Symbols to Continuous Representations

A prerequisite for applying machine learning to textual data is a numerical representation of language. Early approaches relied on discrete, sparse representations such as bag-of-words vectors or TF-IDF weighting, which discard word order and cannot capture semantic similarity between terms. The introduction of distributed word embeddings [91] marked a paradigm shift: words are mapped to dense vectors in a continuous space such that geometric proximity reflects semantic relatedness. This principle extends to subword units, sentences, and entire documents, and it remains the representational foundation for all models discussed in this thesis.

A key limitation of static embeddings is that each word receives a single vector regardless of context. The word *bank*, for instance, has the same representation whether it refers to a financial institution or a river bank. Contextual language models, discussed next, resolve this limitation by conditioning representations on the surrounding text.

2.1.2 The Transformer Architecture

The transformer [350] introduced a sequence modelling architecture based entirely on attention, replacing the recurrent processing of earlier models (e.g., LSTMs) with parallel computation over all positions in a sequence. Its core mechanism is scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V are matrices of queries, keys, and values derived from the input, and d_k is the dimensionality of the key vectors. Multi-head attention

applies this operation in parallel across h independently parameterised subspaces and concatenates the results, allowing the model to attend to different types of relationships simultaneously.

Combined with position-wise feed-forward layers, residual connections, and positional encodings, the transformer achieves three properties that are critical for the tasks in this thesis:

1. **Long-range dependency modelling.** Attention connects every pair of positions in a sequence directly, enabling the capture of dependencies that span hundreds of tokens. This is essential for processing long clinical documents (Contributions P7, P8, P13) and full-length research articles (Contributions P1 through P4).

2. **Bidirectional context.** Encoder-based transformers such as BERT [96] process text in both directions, producing representations that are conditioned on the full surrounding context. Several contributions in this thesis (P5, P6, P7) rely on BERT-derived representations for classification and disambiguation.

3. **Parallelism and scalability.** Unlike recurrent models, transformers process all positions simultaneously, enabling efficient training on large corpora. This property underlies the pre-training paradigm discussed below and is a precondition for the large language models addressed in Theme 5.

Figure 2 illustrates the key components of the transformer architecture.

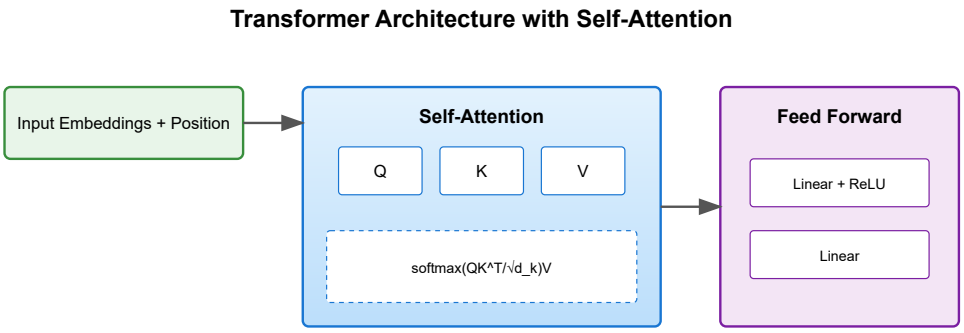


Figure 2: Transformer architecture: input embeddings with positional encoding, multi-head self-attention, and position-wise feed-forward networks.

2.1.3 The Pre-train / Fine-tune Paradigm

Modern NLP follows a two-stage learning paradigm. In the first stage, a model is *pre-trained* on a large, general-purpose corpus using self-supervised objectives such as masked language modelling (BERT) or autoregressive

next-token prediction (GPT, LLaMA). This stage equips the model with broad linguistic knowledge, including syntax, semantics, and world knowledge encoded in its parameters.

In the second stage, the pre-trained model is *fine-tuned* on a smaller, task-specific dataset, adapting its representations to the target domain. This paradigm is remarkably effective in low-resource settings because the pre-trained representations transfer well across tasks and domains [96, 300].

The contributions in this thesis exploit this paradigm in three distinct ways:

1. **Feature extraction.** Pre-trained models are used as fixed or lightly adapted feature extractors whose representations feed into task-specific architectures. This strategy is employed in Contributions P5 and P6 (author disambiguation), P7 (ICD coding), and P3 and P4 (multimodal metadata extraction).

2. **Direct fine-tuning.** The entire pre-trained model is fine-tuned end-to-end on the target task. Contribution P8 [56] fine-tunes LLaMA for ICD code classification, and Contribution P13 [137] fine-tunes LLMs of varying sizes for clinical information extraction.

3. **Reinforcement learning from human feedback (RLHF).** A fine-tuned model is further optimised using reward signals derived from human preferences. Contribution P11 [192] extends this to multi-objective reinforcement learning, balancing several reward criteria simultaneously. The theoretical details of RLHF and multi-objective optimisation are presented in Section 2.6.

2.1.4 Sequence Labelling

Several contributions in this thesis frame their task as sequence labelling, where each token in an input sequence is assigned a label from a predefined set (e.g., `B-Author`, `I-Title`, `O`). Two architectures are particularly relevant.

Conditional Random Fields (CRFs). CRFs model the conditional probability of an entire label sequence $\mathbf{y}$ given an observation sequence $\mathbf{x}$, capturing dependencies between adjacent labels. Contribution P1 [46] and the EXCITE toolchain (P15 [158]) use CRF-based models for reference extraction, where enforcing label transition constraints (e.g., an `I-Author` tag can only follow a `B-Author` tag) significantly improves extraction quality.

Neural sequence labelling. Later contributions replace or augment CRFs with neural architectures. Contribution P2 [49] uses convolutional neural networks with transfer learning. Contributions P3 and P4 introduce multimodal encoders that combine textual token embeddings with visual features

derived from the document image. In all cases, the final prediction layer assigns a label per token, but the underlying feature representations become progressively richer across the contribution arc.

2.1.5 Evaluation Methodology

The contributions in this thesis are evaluated using standard metrics appropriate to each task type. Classification and extraction tasks use precision, recall, and F_1 score at the token or entity level. Time series prediction (P9) uses mean squared error, mean absolute error, and directional accuracy. Disambiguation tasks (P5, P6) use clustering metrics including pairwise F_1 and B^3 measures. Model compression (P12) reports both task performance and computational cost reduction. Task-specific evaluation details are provided in the respective theme sections and in the individual contributions themselves.

The foundations introduced in this subsection, namely continuous text representations, transformer-based architectures, the pre-train / fine-tune paradigm, and sequence labelling, recur throughout the remainder of this thesis. The following subsections build on these foundations to address the specific challenges of each research theme.

2.2 Scholarly Knowledge Infrastructure

Problem and motivation. The FAIR principles [361] require that scholarly data be findable, accessible, interoperable, and reusable. A prerequisite for meeting these criteria is the availability of structured metadata, yet such metadata remains absent for a large fraction of the scholarly record. In the EXCITE project[4], approximately 56% of articles in German social sciences published before 2016 lacked structured metadata. European and German initiatives such as EOSC[5] and NFDI[6] are investing heavily in research data infrastructures, but scalable automated methods are essential to process the existing backlog of unstructured scholarly documents.

Beyond metadata extraction, two further challenges arise. First, analytical methods must be applicable across institutional boundaries where raw data cannot be shared due to privacy or governance constraints. Second, once metadata has been extracted, its quality and FAIR compliance must be assessed, ideally in an automated and interpretable manner.

Methodological landscape. Metadata extraction from scholarly documents has progressed from rule-based and template-driven systems through CRF-based sequence labelling (e.g., ParsCit [80], CERMINE [341]) to deep

[4] https://excite.informatik.uni-stuttgart.de/

[5] https://eosc.eu/

[6] https://www.nfdi.de/

learning approaches (e.g., GROBID [227] and neural variants [285, 16]). A recent line of work integrates visual layout features with textual representations for document understanding [52, 57]. For topic modelling, foundational work on LDA [41] and its hyperparameter sensitivity [191] informs representation choices across scholarly analytics. A detailed review of related work for metadata extraction is provided in Contributions P1 through P4; for topic modelling in P14.

Gaps and contributions. This thesis addresses the scholarly knowledge infrastructure challenge at three levels. At the *extraction* level, Contributions P1 [46] through P4 [57] trace a methodological arc from CRF-based reference extraction to multimodal spatial-semantic frameworks, demonstrating consistent gains from progressively richer representations. These methods are operationalised in the EXCITE toolchain (P15 [158]), which provides end-to-end reference extraction, matching, and publishing at scale.

At the *analysis* level, Contribution P14 [191] provides a principled framework for understanding topic model behaviour through Rényi entropy, while Contribution P16 [51] addresses the institutional barrier by presenting PADME-SoSci, a platform for privacy-preserving distributed machine learning that enables cross-site model training without data centralisation.

At the *assessment* level, Contribution P17 [314] closes the loop with an LLM-based approach for automated FAIRness assessment of metadata. Together, these contributions form a coherent pipeline from extraction through analysis to quality assessment, addressing the scholarly knowledge lifecycle end to end.

The metadata produced by Theme 1 provides the foundation for the identity resolution task addressed in Theme 2.

2.3 Author Name Disambiguation

Problem and motivation. Accurate metadata is necessary but not sufficient for reliable scholarly analytics; the identity of the authors behind each record must also be resolved. Author name disambiguation is complicated by the prevalence of shared names, variant spellings, and the limited distinguishing information in bibliographic records [320, 113]. Errors propagate into citation counts, h-index calculations, and collaboration network analyses, directly affecting how research impact is evaluated.

Methodological landscape. The field has progressed from feature-engineered classifiers [141] through probabilistic and graph-based models [337, 109] to network embedding methods [384] and deep learning on textual metadata [345]. Comprehensive surveys are provided by Ferreira et al. [114] and Sanyal et al. [306]. A detailed review of existing approaches is given in the related work sections of Contributions P5 and P6.

Gaps and contributions. Most deep learning approaches for disambiguation relied on rich metadata (abstracts, affiliations, citation links) that is not always available. Whether minimal metadata alone, specifically short title strings and co-authorship information, suffices for accurate disambiguation with modern representation learning had not been systematically investigated. Contributions P5 [47] and P6 [48] address this gap by training deep neural networks on minimal bibliographic metadata from DBLP, demonstrating that even limited information carries sufficient signal when processed with appropriate representation learning. P5 received the Best Paper Award at TPDL 2022, and P6 extends the evaluation to larger-scale DBLP data with a detailed analysis of feature contributions across ambiguity levels.

These contributions demonstrate that the metadata extraction advances of Theme 1 directly enable downstream scholarly analytics, and the representation learning strategies developed here inform the classification approaches of Theme 3.

2.4 Domain-Specific Text Classification and Prediction

Problem and motivation. A recurring challenge across application domains is that domain texts are long, terminologically dense, and associated with large, imbalanced label spaces. This thesis addresses two instances.

In healthcare, automatic ICD coding requires mapping clinical discharge summaries to a taxonomy of thousands of diagnosis and procedure codes. The task is complicated by extreme code sparsity (many codes appear rarely), overlapping medical terminology across conditions, and the length of clinical documents [247, 279]. Automating this process is critical because manual coding is time-consuming, error-prone, and requires specialised medical expertise.

In finance, public sentiment expressed on social media can significantly influence cryptocurrency market dynamics [197]. Understanding how textual signals interact with traditional trading indicators to drive volatility requires methods that integrate heterogeneous data sources across different temporal scales.

Methodological landscape. For ICD coding, approaches have evolved from rule-based systems [279] through CNN-based architectures with label-wise attention [247, 212] to transformer-based models [40] and knowledge-enriched architectures [371, 28]. For financial sentiment analysis, domain-adapted language models such as FinBERT [21] and multimodal forecasting frameworks combining textual and numerical signals have shown promise [44, 374]. Detailed reviews of related work are provided in Contributions P7 and P8 (ICD coding), P9 (cryptocurrency forecasting), and P13 (clinical information extraction).

Gaps and contributions. In healthcare, two gaps motivated the contributions in this thesis. First, the sparsity of ICD codes and the limited size of labelled clinical datasets required architectures that can leverage external knowledge to compensate for insufficient training examples. Contribution P7 [54] addresses this with a knowledge-guided residual convolutional architecture that enriches text representations with external medical knowledge. Second, the potential of LLMs for clinical text classification, both as direct classifiers and as representation generators for downstream architectures, was underexplored. Contribution P8 [56] investigates this using LLaMA, showing that LLM-derived representations substantially improve classification precision. Contribution P13 [137] complements this by demonstrating that smaller, domain-specifically fine-tuned LLMs can match or exceed larger models in resource-constrained clinical settings.

In finance, Contribution P9 [53] addresses the gap between trading-data-only models and sentiment-aware models by analysing the influence of public awareness and trader sentiment on cryptocurrency volatility, showing that textual features provide significant predictive power beyond traditional market variables.

The common methodological thread across both domains is the *integration of heterogeneous knowledge sources* to compensate for limitations of the primary textual data. The LLMs used as tools in this theme become themselves the object of study in Theme 5.

2.5 Multimodal Interpretability

Problem and motivation. As deep learning models grow in complexity, understanding why a model produces a given prediction becomes essential for trust, accountability, and scientific insight. This is particularly acute in multimodal settings such as Visual Question Answering (VQA), where a model must jointly process an image and a natural language question to produce an answer [19, 128]. The opacity of these models limits their adoption in high-stakes applications and hinders error diagnosis.

Methodological landscape. Interpretability methods for deep learning range from gradient-based saliency maps [311, 319] and surrogate models [299] to counterfactual explanations [129, 351]. Counterfactual approaches are particularly compelling because they answer the question "what minimal change to the input would change the output?", providing actionable and intuitive explanations. In VQA, counterfactual methods have been applied to both the question side [73] and the image side [129]. A detailed review is provided in Contribution P10.

Gaps and contributions. At the time of this work, counterfactual image generation methods for VQA interpretation were limited in their ability

to produce realistic minimal modifications that change a model's prediction while preserving the overall image coherence. Contribution P10 [55] introduces COIN, a counterfactual image generation method that identifies which image regions are decisive for a VQA model's prediction by producing minimal image modifications that flip the predicted answer. COIN operates without requiring access to model internals, making it applicable as a post-hoc explanation method for any VQA architecture.

The interpretability concerns addressed here connect to the multimodal fusion approaches in Theme 1 (P3, P4) and to the broader question of model transparency that motivates the optimisation work in Theme 5.

2.6 Large Language Model Optimisation

Problem and motivation. Several contributions in this thesis use LLMs as components of their pipelines: P8 and P13 for clinical text classification, P17 for FAIR assessment. However, the computational cost of training and deploying these models poses practical barriers, particularly in resource-constrained settings such as clinical environments or smaller research institutions. Two complementary challenges arise: efficiently fine-tuning LLMs when multiple competing objectives must be balanced, and compressing trained models for deployment without unacceptable performance loss.

Methodological landscape. For fine-tuning, Reinforcement Learning from Human Feedback (RLHF) [265] has become the dominant paradigm for aligning LLMs with human preferences, typically using proximal policy optimisation [310] or direct preference optimisation [292]. Extending RLHF to multiple objectives simultaneously is an active research area, with approaches ranging from scalarised reward combination to Pareto-frontier methods [145]. For model compression, knowledge distillation [154] transfers knowledge from large teacher models to smaller students, with notable applications including DistilBERT [305], TinyBERT [176], and recent LLM-specific approaches [190]. Detailed reviews are provided in Contributions P11 and P12.

Key formalisms. Unlike the preceding themes, the contributions in Theme 5 introduce novel optimisation frameworks that benefit from formal presentation. Contribution P11 [192] proposes EMORL, which addresses multi-objective fine-tuning through ensemble-based hidden-state aggregation. Individual policies $\{\pi_{\theta_i}\}_{i=1}^{n}$ are trained for each objective r_i, and the ensemble policy combines their hidden states:

$$\mathbf{h}_{\text{ensemble}} = \sum_{i=1}^{n} \alpha_i \mathbf{h}_{\theta_i}(x) \tag{2}$$

where $\mathbf{h}_{\theta_i}(x)$ is the hidden state from model i and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ are learned aggregation weights satisfying $\sum_{i=1}^{n} \alpha_i = 1$, $\alpha_i \geq 0$.

Contribution P12 [87] investigates knowledge distillation for LLM compression on question-answering tasks using a composite loss:

$$\mathcal{L}_{\text{KD}} = \alpha \mathcal{L}_{\text{hard}} + (1 - \alpha) T^2 \mathcal{L}_{\text{soft}} \tag{3}$$

where $\mathcal{L}_{\text{hard}}$ is the cross-entropy loss with ground-truth labels, $\mathcal{L}_{\text{soft}}$ is the KL divergence between temperature-scaled teacher and student distributions, and T controls the smoothing of probability distributions.

Gaps and contributions. For multi-objective fine-tuning, existing approaches either required separate training runs per objective (computationally expensive) or used fixed scalarisation weights (inflexible). EMORL (P11) addresses both limitations through ensemble aggregation with learned weights, achieving Pareto-optimal training without requiring separate runs. For model compression, the trade-off between compression ratio and downstream task performance for LLMs on question-answering tasks had not been systematically characterised. P12 provides this analysis, showing that student models retain over 90% of teacher performance while reducing computational requirements by up to 57%.

These contributions address the deployment gap that is directly felt in the clinical and scholarly settings of Themes 1 through 3, closing the loop of the research programme.

2.7 Cross-Theme Synthesis

The five themes presented above are not isolated research threads but are connected by recurring methodological patterns that constitute the coherence of this research programme. Four such patterns merit explicit discussion.

Progressive representation enrichment. Across themes, a consistent finding is that richer input representations yield better task performance. In Theme 1, metadata extraction improves as representations progress from token-level CRF features (P1) through CNN-based transfer learning (P2) to multimodal spatial-semantic encoders (P3, P4). In Theme 2, disambiguation accuracy improves when title strings are processed with contextual embeddings rather than surface-level features (P5, P6). In Theme 3, ICD coding performance improves when text representations are enriched with external medical knowledge (P7) or replaced by LLM-derived embeddings (P8). This pattern suggests that representation quality is a more decisive factor than architectural complexity across all domains studied.

External knowledge integration. A second recurring thread is the integration of knowledge sources beyond the primary input text. Theme 1 integrates document layout information (P3, P4). Theme 2 leverages co-authorship graphs (P5, P6). Theme 3 incorporates medical ontologies (P7), LLM-generated representations (P8, P13), and social media sentiment signals (P9). Theme 4 uses counterfactual image modifications as an external signal for model interpretation (P10). In each case, the heterogeneous knowledge source compensates for limitations of the primary textual data, and the specific integration strategy is adapted to the domain.

From tool user to tool builder. LLMs appear in this thesis in two roles. In Themes 1 and 3, they serve as tools: P8 uses LLaMA for ICD classification, P13 fine-tunes LLMs for clinical extraction, and P17 employs LLMs for FAIR assessment. In Theme 5, LLMs become the object of study: P11 optimises their training and P12 compresses them for deployment. This progression reflects a broader trend in the field and demonstrates that practical experience as an LLM user informs the identification of optimisation challenges.

Modality fusion. Finally, several contributions demonstrate that combining information from different modalities yields improvements over unimodal approaches. Theme 1 fuses text and document layout (P3, P4). Theme 3 integrates textual sentiment with numerical market indicators (P9). Theme 4 jointly processes images and natural language (P10). The consistent benefit of multimodal fusion across these diverse settings supports the central thesis argument that robust knowledge processing requires modality-aware methods.

These cross-theme connections demonstrate that the seventeen contributions of this thesis form a coherent research programme rather than a collection of independent studies. The methodological insights transfer across domains, and the themes build upon one another in the progression from data extraction (Theme 1) through identity resolution (Theme 2) and domain-specific analysis (Theme 3) to model interpretation (Theme 4) and optimisation (Theme 5).

3 Paper 1: Reference Extraction from PDF Documents

An End-to-End Approach for Extracting and Segmenting High-Variance References from PDF Documents

Zeyd Boukhers(✉), *Shriharsh Ambhore, Steffen Staab*

(DOI: 10.1109/JCDL.2019.00035)

Abstract This paper addresses the problem of extracting and segmenting references from PDF documents. The novelty of the presented approach lies in its capability to discover highly varying references mainly in terms of content, length and location in the document. Unlike existing works, the proposed method does not follow the classical pipeline that consists of sequential phases. It rather learns the different characteristics of references to be used in a coherent scheme that reduces error accumulation by following a probabilistic approach. Contrary to conventional references, mentioning the sources of information in some publications, such as those of social science, is not subject to the same specifications such as being located in a unique reference section. Therefore, the proposed method aims to extract references of highly varying reference characteristics by relaxing the restrictions of existing methods. Additionally, we present in this paper a new challenging dataset of annotated references in German social science publications. The main purpose of this work is to serve the indexation of missing references by extracting them from challenging publications such as those of German social science. The effectiveness of the presented methods in terms of both extraction and segmentation is evaluated on different datasets, including the German social science set.

Keywords: *Reference Extraction, Reference Segmentation, Conditional Random Fields, Random Forest*

3.1 Introduction

Acknowledging the scientific contribution of previous research work is necessary to ensure a smooth evolution in scientific fields. Over time, authors have adopted a manner to indicate the sources and the contributions of their fellows by mentioning their names, titles of their publications, etc. Several reasons, among which literature search and recommendation, necessitate making these references available and linked to their citations in a network. Therefore, different techniques have been developed to automatically detect, extract and segment these references[342, 227, 286].

In general cases, reference extraction goes through three steps: 1) *section identification*, 2) *reference extraction* and 3) *reference segmentation*. Section

identification is the process of recognizing the section containing all and only references, where reference extraction is dedicated to extracting individual reference strings from the formerly identified reference section. Reference segmentation is used afterwards to segment these references into components (e.g. author, title, volume, etc.). To the extent of our knowledge, researchers have addressed these problems separately by using probabilistic approaches, mainly the Hidden Markov Model (HMM) [152] and Conditional Random Fields (CRF) [275] since the processes satisfy the Markov property. Due to the aperiodicity of some states for both extraction and segmentation tasks, CRF has been widely and could achieve satisfactory accuracy. More specifically, references are assumed to appear in one section, while each reference string has a unique title, a unique page range and a unique source. Also, Neural Networks are also employed to segment reference strings [286], by replacing the words with their numerical representation as an input of the CRF model. Considering the correlation between the two steps, the result of segmentation highly depends on the result of the extraction. In this regard, any deficiency in the extraction's performance will certainly negatively influence the entire process.

Although the citation style is relatively standard in terms of reference content, the citation practice differs from one community to another. Some disciplines, like German social sciences or humanities, commonly use word processors without tool support to represent references, leading to a large variety of reference characteristics within and between publications. Variations include the locations of references in footnotes, endnotes or in specific sections as well as the manner in which reference components are delineated from each other. This variation has different facets, making the automatic recognition of references more challenging. Regarding reference extraction, the main variation is the section in which references are located. Contrary to the standard practice, where there is a single section containing references, it is very common in some communities that the references are placed in separate parts of the publication such as footnotes and endnotes. Here, it is difficult to adapt it to the problem of estimating a sequence of hidden variables. Another facet of this variety is the way of listing the references, which can be either in numerical order, by the full first author's name and the year or only by the first few letters of the name.

Furthermore, there are other varieties within the references being cited in one publication, where references are not composed of the same number of components. For example, social scientists often cite grey literature publications, which do not have author names or page ranges. Consequently, the order of components changes from one reference to another depending on the existence of these components. This leads to making the number of possible transitions higher with fewer examples and hence an imprecise estimation for small datasets.

In order to overcome the above-mentioned issues, this paper proposes a

robust approach to extract and segment difficult references, with a case study of German social science publications. For this, each line in the publication is classified, using Random Forest, into a reference or not. Each classified line is associated with classification scores for all classes (non-, first, intermediate or last reference line). Next, CRF is applied on potential reference lines, with the support of a probabilistic approach, inspired by Metropolis-Hasting (MH), to form complete segmented reference strings. A feedback mechanism is adopted to reduce the accumulation of error among steps, where the approach does not independently rely on each model. Instead, it incorporates the trained models under a probabilistic approach to enhance the results of all models,

The remainder of this paper is structured as follows: Section 3.2 discusses the related and prior work. Section 3.3 introduces the proposed approach starting with feature extraction, the core of the method and the filtering process. Section 3.4 describes the performed experiments and reports the study results of different evaluations. Finally, Section 3.5 concludes and communicates findings of the benefit of the proposed method.

3.2 Related Work

A prior literature review demonstrates that much more effort has been devoted to developing techniques for reference segmentation than for reference extraction. This section discusses the novelties of our method by elaborating on prior work about both reference extraction and reference segmentation.

3.2.1 Reference Extraction

Reference extraction refers to the task of recognizing a section containing reference strings and identifying them afterward. In many disciplines like computer science or mathematics, the reference section is most often clearly delineated from the main text and labeled with a title like 'References' or 'Literature'. Based on this assumption, most prior work recognizes the beginning and end of such a reference section using rule-based or machine learning-based techniques.

Zou *et al.* [400] employ Support Vector Machine (SVM) to locate the reference section in HTML medical articles. They extracted geometric and text features from paper zones and combine them to distinguish between sections. Due to its efficiency, Tkaczyk *et al.* [342] use SVM to predefine a number of frequently occurring document segments, including 'abstract', 'body', 'references' and 'appendix'. Then, they extract the references from only the reference segment. Instead of SVM, the approaches proposed by Patrice Lopez [227] and Körner *it al.* [194] use CRF to extract reference strings in view of its capability to model decision boundaries among different classes. A popular reference extracting tool, called *ParsCit* [80], uses a set

of heuristics to identify the references by scanning the entire document for section headers such as "Reference", "Bibliography", "Notes" or any possible variations.

Considering the difficulties of identifying the sections in PDF documents, Bergmark [35] first converted the document into a well-formed XML format and then parsed it to find the section labeled with "References". Another way to identify the section containing references is a rule-based approach [43], which tends to achieve better results as the rules are customized for a specific domain. However, taking into account the differences among reference styles and articles in general, the rule-based techniques do not perform well on articles for which rules were not priorly defined.

3.2.2 Reference Segmentation

In the literature, many methods for reference string segmentation (also known as reference string parsing) exist, differing in their techniques [340], assumptions and target datasets. We broadly classify these methods into two categories, namely Classifier-based and Template-based.

Classifier-based Techniques: Machine Learning (ML) algorithms are popular among researchers of information extraction as they are capable to learn from different data and achieve higher performance. Supervised algorithms such as CRF [275], Hidden Markov Models (HMM) [152] and SVM [390] proved their efficiency to achieve a satisfactory result in reference segmentation. Another direction to segment references adopts unsupervised learning techniques such as Hierarchical and Agglomerative clustering [271, 131], where little or no data is required to train the classifier.

SVM is one of the most frequently used methods for classifying tokens of a given reference string. Zhang *et al.* [390] propose a structural SVM to segment references of biomedical literature. Due to the strong regularity of the reference structure, the task is considered as a sequence learning problem, where the achieved accuracy is about 98%. Zou *et al.* [400] used and compared Support Vector Machine and Conditional Random Fields for reference segmentation, focusing on articles in the medical domain. The comparison concluded that both approaches achieve nearly the same accuracy (97%).

Since segmenting references corresponds to the problem of finding the most likely sequence of hidden states, HMM is a suitable tool for estimating a sequence of hidden variables given the sequence of observed events. In [152], a simple first-order HMM is applied, where the data is smoothed using naive smoothing to handle the absence of some state transitions and emissions. The model was trained on handcrafted citation training data and achieved an accuracy of more than 90% when tested on health science datasets consisting of homogeneous citations. Furthermore, Yin *et al.* [380] used Bigram HMM to solve this problem, claiming that their method yields better results than

Unigram HMM. The difference between Unigram HMM and Bigram HMM is that the latter uses a modified model for computing the emission probability, without changing the structure of HMM itself.

Conditional Random Fields are statistical models used to compute the probability of hidden states given the sequence of observations [203]. Peng and McCallum [275] conducted an empirical study on Gaussian variations, Exponential and Hyperbolic-L1 prior and several class of features. They claimed that their method attained the state-of-the-art performance for extracting standard fields in the citations. Romanello *et al.* [302] used CRF for extracting and segmenting canonical references found in the field of classical studies. The popular citation extraction tool *ParsCit* developed by Councill *et al.* [80] relies on CRF model to identify the labels such as *Author*, *Title* and *Year* for the tokens being observed in the references.

Since the task of building a gold standard dataset is cumbersome and resource intensive, the problem of segmenting references was addressed using unsupervised learning approaches. For example, Grenager *et al.* [131] adopted this approach with small amounts of prior knowledge to segment fields, considering two different datasets, namely bibliographic citations and classified advertisements. They used Unconstrained HMM and Hierarchical Mixture Emission model for evaluating it against a supervised first-order HMM model in terms of average accuracy. Although the accuracy obtained by the unsupervised model was not better than that of the supervised model, the authors assume that the accuracy can further be increased by rectifying the structure of the model.

Semi-supervised learning approach was also explored in [131], where an increase of the model accuracy was noticed by incrementally introducing annotated citation data into the dataset being used to learn the unsupervised model. Chambers and Jurafsky [65] also leveraged unsupervised learning methods for template mining without knowing the templates in advance.

Basically, the performance of a machine learning algorithm depends on the features that represent the data. In the context of extracting metadata from references, many techniques achieved compelling performance using different features, which were manually designed. It is also important to note that these features are engineered for a specific domain and are not necessarily well generalized when applied to another domain or reference style. This deficiency can be overcome with the help of Deep Neural Networks which are considered to be more effective in obtaining an accurate and generalized representation of the data. Prasad *et al.* [286] exploited this approach by employing a Long Short Term Memory (LSTM) neural network model to represent tokens. Afterward, CRF model was trained on the extracted features, where this method showed strong performance over manually defined features.

Template-based Techniques: To extract relevant information, templates are formed based on prior domain knowledge. These templates are directly applied to the text data. Relevant information is extracted when certain conditions included in the template are matched [98]. Yang *at al.* [71] enhanced their previous work on sequence alignment techniques by replacing the reliance on background knowledge (e.g., author name database) with the punctuation symbols to identify the reference format. The authors demonstrate that the improved "BibPro" citation parser significantly performed better than INFOMAP and ParsCit metadata extraction tools if it used with a dataset containing six different citation styles

Also, a Hierarchical template-based approach *INFOMAP* was proposed by Day *et al.* [88]. Considering the disadvantages of traditional rule-based approaches such as lack of customization of the rules, the template-based hierarchical approach overcomes them by representing the information about the reference components (e.g., Author, Title, Volume, and Issue) in a tree structure. This structure represents patterns for the reference components found in different reference styles. The performance of INFOMAP was evaluated on 10,000 reference strings randomly selected from six different reference styles (e.g., IEEE, APA, ACM, etc.), demonstrating an average accuracy of 92.39%.

The template-based techniques require a thorough domain knowledge for designing the templates. Accordingly, Chambers and Jurafsky [65] proposed a method to learn the template structures instead of designing them. For this, the authors relied on two unsupervised learning algorithms: Latent Dirichlet Allocation (LDA) and Agglomerative clustering. The obtained result on the MUC-4 corpus leads to the conclusion that the learned templates perform better than the templates created by domain experts.

3.3 Approach

The proposed method of extracting references from PDF documents operates in two correlated phases: reference line classification and reference segmentation & identification. Before applying the method on query documents, an offline process has to be carried out on the training set to train the necessary models. Subsequently, the online process employs the trained models to extract and segment reference strings from different PDF documents. Fig. 3 illustrates the general overview of the proposed approach consisting of an offline and online process.

To train the models, layout information in text format is automatically extracted from the training PDF documents using Cermine [342]. From the output of each PDF document, the references were manually recognized and annotated, where each reference was in return manually segmented into basic components (e.g., author, title, source, etc.). These references, consolidated with the remaining text of their corresponding PDFs, represent the gold

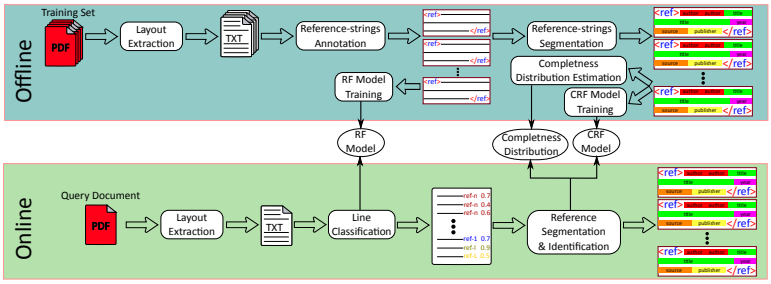


Figure 3: A general overview of the reference extraction approach. The offline process is dedicated to training the classifiers: reference line (Random Forest 'RF') and reference segmentation (Condition Random Fields 'CRF') and estimating the completeness distribution using the manually extracted and segmented references. The online process uses the trained models to extract and segment the references of a query PDF document.

standard of our dataset. Section 3.4 gives detailed information about the dataset.

Using the training set of the gold standard, a Random Forest model is trained for line classification considering four classes: non-, first, intermediate and last reference line. The reason for choosing Random Forest over other classifiers such as SVM is its capability to handle multiple classes and support features on various scales. Besides, the manually segmented reference strings in the gold standard are used to train the CRF model in order to observe the relationships among components. Additionally, the completeness of a reference is estimated as probability density functions of the set of segmented reference strings. Each function is estimated with a Multivariate Kernel Density Estimator given one of the forming properties (i.e., existing components, number of tokens per each component, etc.) of the training dataset. Below the extraction of features, the core of the approach and filtering process are discussed:

3.3.1 Features

Both models, responsible for classifying lines (i.e. Random Forest) and segmenting reference string candidates (i.e. Conditional Random Fields), use a set of discriminative features extracted from each line and each token, respectively. List. 1 presents a brief description of the features used by our method, where a detailed explanation can be found under this repository path[7].

[7]Git repository: `https://github.com/exciteproject/Exparser`

mylist 1: Overview of the considered features in line classification and reference segmentation.

a Format-based

- Existence of year format, e.g., *1999*
- Existence of page format, e.g., *25–32*
- Existence of hyperlink format, e.g., *http://www.xyz.com*
- etc.

b Lexical-based

- Existence of the keyword *Vol.*
- Existence of the keyword *Eds.*
- Existence of the keyword *pp.*
- etc.

c Semantic-based

- Existence of a first or last name, e.g., *Alexander*
- Existence of a city name, e.g., *Paris*
- Existence of a source name, e.g., *International Conference on...*
- etc.

d Shape-based

- Ratio of digits in a Line/Token
- Ratio of capital letters in a Line/Token
- Histogram of word length in a Line
- etc.

Reference Extraction & Segmentation Given a query document, the layout information is similarly extracted using Cermine [342]. Subsequently, the pre-trained Random Forest model is used to classify each line into either: non-, first, intermediate or last reference line (i.e., ref-0, ref-1, ref-I and ref-L, respectively). Here, each classified line is associated with the probabilities of belonging to all classes. The classified lines and the associated probabilities are used afterwards to compose and segment consistent references. A consistent reference denotes a combination of lines that potentially fulfil the conditions of a complete and coherent reference string. In this regards, the method starts with the line $\ell_{\hat{i}}$ having the highest reference probability among all lines (Λ) being classified as a *reference line* (i.e. ref-1, ref-I or

ref-L) , where $\hat{i}$ is obtained as follows:

$$\hat{i} = \underset{\substack{0<i\leq N \\ i\in\Lambda}}{\operatorname{argmax}} P_e\left(\omega^e(\ell_i) \neq \text{ref-0} \mid \boldsymbol{c}_i^e\right) , \tag{4}$$

N is the total number of lines in Λ and ℓ_i denotes the ith line. $\omega^e(.)$ is the line class and $\boldsymbol{c}_i^e$ is the corresponding extracted feature vector. $P_e(. \mid .)$ represents the conditional probability of the line class, given the corresponding feature vector, which is computed by the pre-trained Random Forest model.

The unique selected line is considered as an initial reference-string candidate $\psi_{t=0}$. Then, a series of candidates are sequentially generated in a random process, starting from $\psi_{t=0}$, where the best candidate is assumed to be approached with the progress of this process. Considering that the number of lines composing a reference string is unpredictable, the number of candidates (α) in our experiments is set to 30 to ensure that the best candidate is reached. Let ψ_t and ψ_{t-1} be two consecutive reference-string candidates, the superiority between them is assessed by the acceptance ratio a, which measures the quality of reference-string candidates in terms of line combination (Δ_e), segmentation (Δ_s) and completeness (Δ_c). Here, each candidate is compared to its predecessor, where it is accepted if it is better, otherwise, it is rejected and substituted with the predecessor candidate. In other terms, since the comparison is achieved by computing the acceptance ratio a, the new candidate is accepted only if $a > 1$. The acceptance ratio between the current candidate ψ_t and its predecessor ψ_{t-1} ($a_{t,t-1}$) is computed as follows:

$$a_{t,t-1} = \frac{\Delta_e(\psi_t)}{\Delta_e(\psi_{t-1})} \times \frac{\Delta_s(\psi_t)}{\Delta_s(\psi_{t-1})} \times \frac{\Delta_c(\psi_t)}{\Delta_c(\psi_{t-1})} . \tag{5}$$

Eq.5 approximates the optimal candidate of line combination for each reference by assessing the disparity among the qualities of sampled candidates. It is important to note that the number of samples (α) should be sufficient to obtain a stabilized reference string.

Considering a candidate ψ, the quality measure of line-combination validates the determination of ψ by first-reference line from the top and the last reference line from the bottom. In this context, an adequate reference candidate is characterized by top and bottom lines having high probabilities of first and last reference lines, respectively. The line-combination measure of a given candidate $\Delta_e(\psi)$ is obtained as follows:

$$\begin{aligned} \Delta_e(\psi) = & P_e\left(\omega^e(\psi^1) = \text{ref-1} \mid \boldsymbol{c}^e(\psi^1)\right) \\ & \times P_e\left(\omega^e(\psi^L) = \text{ref-L} \mid \boldsymbol{c}^e(\psi^L)\right) , \end{aligned} \tag{6}$$

where ψ^1 and ψ^L correspond to the first and last line of ψ, respectively.

The segmentation measure (Δ_s) is a precision evaluation of the segmentation that is applied on ψ, and it is computed as follows:

$$\Delta_s(\psi) = \prod_{j}^{M} \max_{0<k\leq K} P_s\left(\omega_k^s \mid \boldsymbol{c}_j^s\right) \ , \tag{7}$$

where M is the number of tokens in ψ, ω^s is the class of reference component (i.e., Author, Title, Year, etc.) and K denotes the number of all possible reference components. $\boldsymbol{c}_j^s$ represents the extracted feature vector from the jth token in the corresponding reference candidate.

As its name indicates, the completeness measure evaluates the completeness of the reference candidate given the references in the training dataset.

Let f_a, f_b and f_c be the estimated functions that represent the distributions of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, respectively, obtained using Multivariate Kernel Density Estimation. $\mathbf{x} = (x_1, x_2, \cdots, x_R)$ represents the existence of key components (i.e. Author name, Year, Page, Source and Editor) in the training dataset of length (R). Similarly, $\mathbf{y} = (y_1, y_2, \cdots, y_R)$ and $\mathbf{z} = (z_1, z_2, \cdots, z_R)$ represent the arrangements of the key components in terms of their first and last appearance, respectively. Furthermore, other heuristic completeness factors (f_d) are used, for example, the length of lines and punctuation marks at the end of lines. A detailed explanation of the completeness factors is found under the corresponding repository link[8].

For measuring the completeness of a reference string candidate ψ, given its set of tokens and their corresponding components, $\Delta_c(\psi_t)$ is computed as the product of $f_a(\psi)$, $f_b(\psi)$, and the other completeness factors.

$$\Delta_c(\psi) = f_a(\psi) \times f_b(\psi) \times f_c(\psi) \times f_d(\psi) \ . \tag{8}$$

For the sake of simplicity, Fig. 4 shows an illustrative example of the computation of quality measures given a reference-string candidate. In the top-left, the line combination measure is obtained as the product of two scores: 1) the probability that the first line of the reference-string candidate =ref-1 and 2) the probability that the last line of the reference-string candidate =ref-L. In the top-right, the products of the completeness densities imply the completeness measure. On the bottom side, the segmentation measure is represented by the product of components' scores.

After determining the first reference string based on the above process, the method iteratively searches for the remaining reference strings until no reference line remains. It starts with the line having the highest probability among the remaining lines and follows the same process to form and segment a new coherent reference string. Algorithm. 1 summarizes the above-discussed steps.

[8]Git repository: `https://github.com/exciteproject/Exparser`

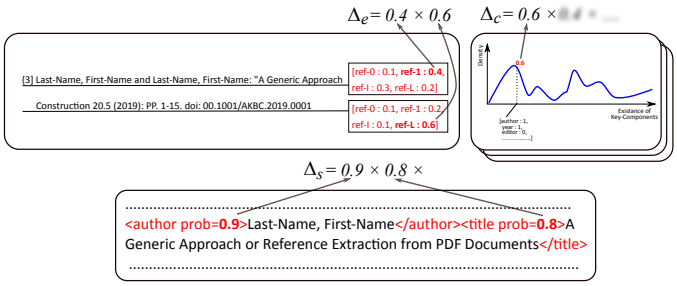


Figure 4: An illustration of the quality measure given a reference candidate.

Algorithm 1 Algorithm of extracting coherent reference strings

$\Theta = \emptyset$ ▷ Set of references
while $\Lambda \neq \emptyset$ **do**
  Initialise with $\ell_{\hat{i}}$ (among $\Lambda_{0:N}$) using Eq. 4
  Determine the initial candidate: $\psi_{t=0} = \ell_{\hat{i}}$
  Compute the measures of $\psi_{t=0}$ following: Eq. 6, Eq. 7 and Eq. 8
  for $t \leftarrow 1$ to α **do** ▷ $\alpha = 30$
    Generate a new candidate $\psi_t = \ell_{x(t):y(t)}$ from ψ_{t-1}, ensuring $\ell_{x(t):y(t)} \subset \Lambda$.
    Compute the measures of ψ_t following: Eq. 6, Eq. 7 and Eq. 8 $\Rightarrow$ obtain a
    if $a < 1$ **then**
      $\psi_t = \psi_{t-1}$
    end if
  end for
  $\Theta = \Theta \cup \psi_\alpha$
  $\Lambda = \Lambda \setminus \ell_{x(\alpha):y(\alpha)}$
end while

A detailed explanation of the method is given in an online documentation [9].

Filtering In the online process, the lines of the query document are independently classified, without considering the classes of subsequent and former lines. Although it is considered an advantage to find references in different parts of the document (e.g., footnote), the classification output is likely noisy with wrong classifications. To overcome this problem, a filtering process is necessary to smooth the output. The reference regions are first searched by employing a capacity distribution that expresses each line with the number of neighbour reference lines. Let ε be the stride parameter, the density d_i of each line ℓ_i is computed as:

[9] `https://exparser.readthedocs.io`

$$d_i = \sum_{\iota=i-\varepsilon}^{i+\varepsilon} h(\ell_\iota), \quad \text{where} \quad h(\ell_\iota) = \begin{cases} 0 & \text{if } \omega^e(\ell_\iota) = \text{ref-0} \\ 1 & \text{Otherwise} \end{cases} . \tag{9}$$

Based on the above equation, the wrongly detected reference lines can be discarded. Also, the wrongly missed reference lines can be reconsidered. Note that this rectification does not affect the scores of classes given by Random Forest, where they remain the same for the rectified lines.

3.4 Experiments

To validate the effectiveness of the proposed method in terms of both reference extraction and segmentation, several experimental evaluations have been carried out on different datasets due to the variety of their properties and characteristics. Another reason to use different datasets is the fact that each dataset is derived from a well-known state-of-the-art method. Therefore, for precise comparison, the proposed approach is compared to *Cermine*, *Grobid* and *ParsCit* on their corresponding datasets. Additionally, this paper proposes a new challenging dataset, on which the proposed method, *Grobid* and *Cermine* are applied and compared. The efficiency of the methods is assessed in terms of precision, recall and F-score, where at the end the complexity of each method is discussed. In the following, the considered datasets are described:

Proposed Dataset in the German language (*PGS*): This dataset consists of 125 annotated articles in the German language collected from SSOAR[10]. All articles are on social science and can be divided into two categories: 1) 100 articles have references in a specific section and 2) the references in the remaining 25 articles are sparsely located in the document and not only in the reference section. Fig 5 illustrates two examples of references and notes in German social science publications. As shown, the similarity between references in footnotes and notes is very high. This includes; 1) similar location in the paper, same font style and size, and 3) enumerated in the same order. On the other hand, a reference in the reference section appears differently in terms of format and content.

The references in each article are manually identified, segmented and consolidated with the remaining text in the layout file. By considering the reference strings in all articles, 2652 reference strings are extracted and assigned to 6711 text lines. Here, the lines are extracted using a tool from *Cermine* that extracts each sequence of words in the PDF with respecting line break and multi-column PDF. It has to be noted that the references in this dataset do not correspond to only academic literature but grey literature as well. Therefore, different unusual references are more likely to appear in

[10] `https://www.gesis.org/ssoar/home/`

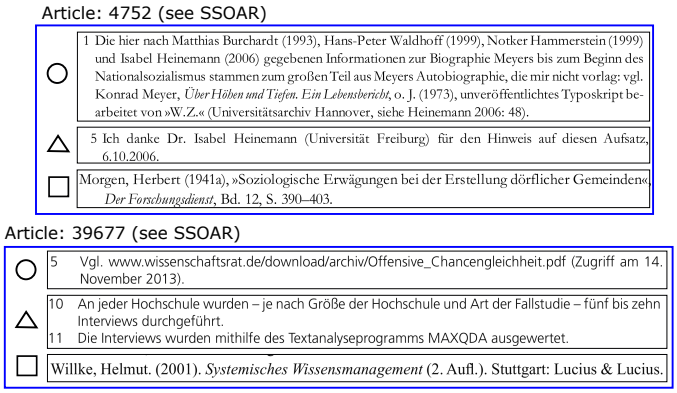


Figure 5: Examples from two different articles showing the difference and similarity between references and footnotes. ◯: reference in footnote, △: footnote (non-reference) and □: a reference in a reference section.

this dataset. This includes; references without authors (e.g. organizational reports), and references with only titles and URLs (e.g. datasets). This variety is considered challenging given 1) the existence and absence of essential components (e.g. author), 2) the non-unified arrangement of reference components (both inter- and intra-articles), and 3) the variety of component numbers.

In the remainder of this section, the parts of extraction and segmentation of *PGS* are referred to by PGS_e and PGS_s, respectively.

Proposed Dataset in the English language (*PES*): Similarly to *PGS*, *PES* is a set of 100 PDF articles collected from SSOAR, where all articles are in the English language. However, the references in this collection are from different languages. After annotating all the references in *PES*, 2838 are counted.

Cermine Dataset (*CDS*): It is a part of GROTOAP2 dataset[11], which consists of 13210 articles collected from medicine domain. Due to the considerable time consuming to train the reference segmentation model of *Cermine*, only 6858 reference strings from GROTOAP2 are considered, each of which is segmented into components.

Grobid Dataset (*GDS*): From 1943 articles available in PMC_sample [12], 100 articles are randomly selected to constitute this dataset, where reference strings in each document are annotated and segmented.

[11] `https://repod.pon.edu.pl/dataset/grotoap2`
[12] `https://grobid.readthedocs.io`

For detailed evaluations, reference identification and reference segmentation are independently discussed below:

3.4.1 Reference Identification

Evaluating the quality of reference identification is subjected to several criteria. First, the retrieval accuracy, which is characterized by 1) the number of relevant reference lines among the retrieved ones and 2) the number of retrieved reference lines among the total amount of existing reference lines in the document. As a reference consists of multiple lines, it might be retrieved, either precisely, missing relevant line(s) or including irrelevant line(s). Therefore, the second criterion is the inner precision within the reference itself. Accordingly, a reference is considered to be precisely identified if all its lines are gathered without including outlier lines.

The qualities of reference identification of the proposed method (*Proposed*), *Grobid* and *Cermine* are assessed by applying each method to PGS_e using 10-fold cross-validation. To ensure equitable learning of all the models, we trained *Grobid* on the level of reference-line segmentation and section segmentation. Here, footnotes containing references were re-annotated as reference sections for both training and testing sets. Considering the identification of each reference line independently, Table 2 presents the results of the three datasets in terms of three evaluation macro averaged metrics; precision, recall and F1-score. Here, each line in the document can belong to either a reference line or a non-reference line. As demonstrated, *Cermine* achieves the highest precision among the three models, where about 77% of the retrieved references are relevant. However, most of the reference lines are not retrieved. On the contrary, the proposed method retrieved almost 90% of the reference lines, which is higher with 20% from the successor approach.

Table 2: Result of independent reference line extraction on PGS_e using *Proposed*, *Grobid* and *Cermine*.

	Precision	Recall	F1-Score
Proposed	0.69	**0.89**	**0.78**
Grobid	0.66	0.69	0.67
Cermine	**0.77**	0.26	0.39

In addition to precisely and accurately identifying reference lines, it is also necessary to consolidate them together to form consistent reference strings. Considering that a reference string is characterized by a first-line (ref-1), intermediate line(s) (ref-I) and last line (ref-L), Table 3 demonstrates the results of the three approaches, where that obtained by *Proposed* is very similar to the result in Table 2. This means that most of the retrieved refer-

ence lines are precisely identified (ref-1, ref-I and ref-L) and thus references are well composed.

Table 3: Result of reference extraction on PGS_e using *Proposed*, *Grobid* and *Cermine*, where M-Aver. denotes Micro-Average.

	Precision				Recall			
	ref-1	*ref-I*	*ref-L*	*M-Aver.*	*ref-1*	*ref-I*	*ref-L*	*M-Aver.*
Proposed	0.73	**0.51**	0.73	**0.67**	**0.84**	**0.84**	**0.86**	**0.84**
Grobid	**0.74**	0.42	**0.74**	0.65	0.56	0.73	0.6	0.62
Cermine	0.63	0.09	0.12	0.31	0.47	0.01	0.01	0.19
F1-Score								
	ref-1	*ref-I*	*ref-L*	*M-Aver.*				
Proposed	**0.78**	**0.64**	**0.79**	**0.74**				
Grobid	0.64	0.53	0.66	0.61				
Cermine	0.54	0.01	0.01	0.22				

3.4.2 Reference Segmentation

The quality of reference segmentation is assessed based on several experimental evaluations. To avoid the influence of former phases on the evaluation of this phase, the references are assumed to be properly extracted for all methods. Hence, the input in this evaluation is a set of references, each of which is segmented into components such as *author*, *title*, *pages*, etc. It is important to note that the compared methods don't consider similar components, for example, *URL* is not recognized by *Cermine* and *Identifier* is considered only by the proposed approach. Moreover, the evaluation is carried out after removing non-alpha-numeric characters from the output of all models as well as the ground truth. Due to the variety of reference styles and for objective evaluation, five datasets are used to compare the *Proposed* to the state-of-the-art methods.

To ensure that all models benefit from all available information, the segmented references in PGS_s and PES_s are prepared according to the corresponding format of each model without changing any of their properties (i.e. spacing, punctuation, capitalisation, etc.). Furthermore, to validate the effectiveness of the proposed approach in segmenting regular reference strings, Table 4 presents the results of our model and three other baseline models: *Grobid*, *Cermine* and *ParsCit* on PES_s. The result of each model was obtained by applying 10-fold cross-validation on the 2838 reference strings, where the data is split based on the 100 blocks of reference strings (i.e. articles). Specifically, for each fold, the reference strings in 90 articles are used to train the models and the reference of the remaining articles are used for

testing. The reason is to avoid having relatively similar reference strings (in terms of style, arrangement and components) in training and testing sets, assuming that the reference strings in the same article follow the same style. Since each of the applied methods admits a different set of components, three average results are computed for different sets of components. E.g., $macro\,Average^1$ is the average of all components associated with the attribute (^{1}).

As the table demonstrates, the average results of *Proposed*, *Grobid* and *Cermine* are considering the seven common components are similar. However, considering the result per each component, *Proposed* achieves a satisfactory result for all components, where the minimum precision, recall and F1-score, without considering *Other*, are $(0.898, 0.764, 0.733)$, respectively. Also, most essential components (i.e. author, title, source, etc.), that can be used to identify articles, are retrieved by *Proposed*.

Moreover, we evaluated the approaches on the segmentation set of PGS_s, where the results of *Proposed*, *Grobid* and *Cermine* are also very similar with higher precision for *Grobid* and higher recall for *Proposed* as can be seen in Table 5.

Since the variety of reference styles is large and the importance of components differs from one community to another, as we observed in the annotation of other datasets, we evaluate *Proposed* considering *CDS* dataset using 10-fold cross-validation. In this evaluation, we split the reference strings into training and testing parts without taking into account the articles citing these references. In addition to *Proposed*, *Cermine* is also applied to this dataset and their results are demonstrated in Table. 6. In contrast to the previous evaluation and although its high precision, the recall of the proposed method is relatively lower. This decrease in performance is explained by the annotation of this dataset, in which a lot of content is not annotated. More precisely, there exist references, in which URLs and information about the editor are present but not annotated. In our approach, any content in the reference which is not annotated is considered as either *Other* or *Empty*. *Other* is assigned to content that might be useful such as note and place of publication and its goal is to help the prediction of neighbouring components by exploring the transitions among states. *Empty* is any content that is not useful as information but it helps the understanding of the edges between components by CRF. Examples of *Empty* include: punctuation, keywords (e.g. 'In:'), and parentheses.

Table 4: Result of reference segmentation on *PES* using: P: *Proposed*, G: *Grobid*, C: *Cermine* and R: *ParsCit*

	Precision				Recall				F1-Score			
	P	*G*	*C*	*R*	*P*	*G*	*C*	*R*	*P*	*G*	*C*	*R*
Publisher[1,2,3]	0.959	**0.964**	0.773	0.921	0.845	**0.877**	0.58	0.528	0.897	**0.917**	0.611	0.665
First Page[1,2,3]	**0.997**	0.988	0.982	0.908	**0.98**	0.963	0.972	0.014	**0.989**	0.976	0.977	0.027
Last Page[2]	0.994	0.917	**0.996**	N/A	**0.984**	0.906	0.975	N/A	**0.989**	0.911	0.985	N/A
Title[1,2,3]	0.932	**0.952**	0.829	0.787	**0.973**	0.958	0.958	0.908	0.951	**0.955**	0.888	0.843
URL[3]	**0.965**	0.944	N/A	0.865	0.764	**0.849**	N/A	0.445	0.809	**0.868**	N/A	0.564
Author[1,3]	**0.971**	0.891	0.946	0.963	**0.91**	0.899	0.9894	0.884	**0.938**	0.894	0.918	0.921
Author Surname[2]	**0.952**	0.891	0.889	N/A	0.884	**0.909**	0.873	N/A	**0.915**	0.899	0.88	N/A
Author Given-name[2]	**0.941**	0.79	0.938	N/A	**0.912**	0.855	0.849	N/A	**0.925**	0.821	0.887	N/A
Volume[1,2,3]	0.956	**0.992**	0.957	0.971	**0.937**	0.925	0.928	0.242	0.925	**0.957**	0.942	0.374
Source[1,2,3]	**0.943**	0.941	0.903	0.698	**0.835**	0.832	0.641	0.469	**0.884**	0.883	0.745	0.558
Editor[3]	0.898	**0.906**	N/A	0.498	**0.778**	0.494	N/A	0.12	**0.832**	0.638	N/A	0.19
Identifier	**0.96**	N/A	N/A	N/A	**0.701**	N/A	N/A	N/A	**0.733**	N/A	N/A	N/A
Year[1,2,3]	0.944	**0.991**	0.972	0.96	0.933	**0.95**	0.946	0.766	0.939	**0.97**	0.958	0.85
Issue[2]	0.958	**0.981**	0.978	N/A	**0.889**	0.781	0.846	N/A	**0.922**	0.867	0.906	N/A
Other[3]	**0.846**	0.783	N/A	0.485	0.722	**0.78**	N/A	0.775	0.777	**0.778**	N/A	0.595
macro Average[1]	0.957	**0.96**	0.91	0.887	**0.916**	0.915	0.859	0.544	0.932	**0.936**	0.868	0.605
macro Average[2]	**0.958**	0.941	0.922	N/A	**0.917**	0.896	0.857	N/A	**0.934**	0.916	0.878	N/A
macro Average[3]	**0.941**	0.935	N/A	0.806	**0.868**	0.858	N/A	0.515	**0.894**	0.884	N/A	0.559

Table 5: Result of reference segmentation on PGS_s using: R: *Proposed*, G: *Grobid*, C: *Cermine* and R: *ParsCit*

	Precision				Recall				F1-Score			
	P	*G*	*C*	*R*	*P*	*G*	*C*	*R*	*P*	*G*	*C*	*R*
Publisher[1,2,3]	0.964	**0.97**	.966	0.747	0.811	**0.814**	0.705	0.384	0.875	**0.878**	0.765	0.496
First Page[1,2,3]	0.979	**0.989**	0.957	0.934	**0.938**	0.846	0.887	0.04	**0.958**	0.91	0.919	0.079
Last Page[2]	0.991	0.988	**0.995**	N/A	**0.962**	0.946	0.95	N/A	**0.976**	0.966	0.972	N/A
Title[1,2,3]	0.894	**0.921**	0.817	0.653	**0.961**	0.937	0.921	0.866	0.925	**0.929**	0.865	0.743
URL[3]	**0.996**	0.977	N/A	0.985	0.8	**0.961**	N/A	0.687	0.881	**0.969**	N/A	0.783
Author[1,3]	0.926	0.809	0.882	**0.945**	0.793	**0.884**	0.857	0.732	0.854	0.844	**0.867**	0.824
Author Surname[2]	**0.91**	0.843	0.782	N/A	0.787	**0.881**	0.821	N/A	0.843	0.**861**	0.799	N/A
Author Given-name[2]	**0.89**	0.778	0.887	N/A	0.823	**0.856**	0.791	N/A	**0.855**	0.813	0.828	N/A
Volume[1,2,3]	0.932	**0.988**	0.872	0.926	**0.78**	0.748	0.757	0.144	**0.848**	**0.848**	0.808	0.245
Source[1,2,3]	0.89	**0.898**	0.784	0.488	**0.749**	0.746	0.542	0.487	0.81	**0.814**	0.636	0.484
Editor[3]	0.878	**0.898**	N/A	0.596	**0.751**	0.489	N/A	0.029	**0.808**	0.631	N/A	0.048
Identifier	**0.902**	N/A	N/A	N/A	**0.706**	N/A	N/A	N/A	**0.754**	N/A	N/A	N/A
Year[1,2,3]	0.904	0.977	0.933	**0.98**	0.901	0.907	**0.92**	0.529	0.903	**0.941**	0.926	0.684
Issue[2]	0.964	0.979	**0.99**	N/A	**0.703**	0.574	0.521	N/A	**0.799**	0.715	0.658	N/A
Other[3]	**0.848**	0.695	N/A	0.438	0.735	**0.78**	N/A	0.721	**0.785**	0.73	N/A	0.54
macro Average[1]	0.927	**0.936**	0.887	0.81	**0.848**	0.84	0.798	0.455	**0.882**	0.881	0.827	0.508
macro Average[2]	0.927	**0.936**	0.898	N/A	**0.841**	0.825	0.781	N/A	**0.879**	0.867	0.818	N/A
macro Average[3]	**0.921**	0.912	N/A	0.769	**0.822**	0.812	N/A	0.462	**0.865**	0.849	N/A	0.493

Table 6: Result of reference segmentation on *CDS* using: P: *Proposed* and C: *Cermine*.

	Precision		Recall		F1-Score	
	P	*C*	*P*	*C*	*P*	*C*
Publisher	**0.94**	0.93	0.35	**0.61**	0.43	**0.74**
First Page	0.98	**0.99**	0.96	**0.98**	0.97	**0.98**
Last Page	0.997	**0.99**	0.93	**0.99**	0.96	**0.99**
Title	**0.91**	0.83	0.96	**0.97**	**0.93**	0.89
A.Surname	**0.88**	0.8	0.24	**0.92**	0.38	**0.86**
A.Given-name	**0.98**	0.89	0.16	**0.9**	0.27	**0.892**
Volume	**0.99**	**0.99**	0.91	**0.97**	0.95	**0.98**
Source	0.92	**0.93**	**0.96**	0.61	**0.94**	0.74
Year	0.95	**0.99**	0.91	**0.97**	0.93	**0.98**
Issue	0.95	**0.98**	0.8	**0.86**	0.87	**0.92**
macro Average	**0.95**	0.93	0.72	**0.88**	0.76	**0.89**

As the previous evaluation showed the negative impact of non-annotated content in the prediction of some components, we aim in this evaluation to examine the influence of *Other* and *Empty* on guiding the prediction. For this, we used *GDS* dataset, in which the contents of the annotated reference strings are reordered compared to the raw data. In addition, these annotated reference strings (ground truth) are filtered from additional content, including punctuation and key-words. Since both models *Proposed* and *Grobid* rely on the sequential property of reference strings, we preferred to test both methods on the ground truth after removing the annotation so that we ensure that the training and testing phases are consistent.

The result presented in Table 7 shows a high average precision of the proposed method without relying on additional content. It shows also that the relatively low recall is due to two components: *Editor* and *URL*. For *Editor*, the reason is the lack of instances assigned to this component and the remarkable presence of *Other* which is employed in this case to identify 'place of publication'.

Table 7: Result of reference segmentation on *GDS* using: P: *Proposed* and G: *Grobid*.

	Precision		Recall		F1-Score	
	P	*G*	*P*	*G*	*P*	*G*
Publisher	**0.93**	0.90	**0.77**	0.72	**0.83**	0.79
First Page	**0.99**	0.98	**0.98**	0.93	**0.9**	0.95
Last Page	**0.98**	0.92	**0.97**	0.92	**0.98**	0.92
Title	**0.99**	0.98	**0.99**	0.98	**0.99**	0.98
URL	**0.99**	0.98	0.38	**0.92**	0.39	**0.95**
A.Surname	**0.99**	0.8	**0.99**	0.95	**0.99**	0.87
A.Given-names	**0.99**	0.33	**0.99**	0.49	**0.99**	0.394
Volume	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
Source	**0.96**	0.95	0.91	**0.93**	**0.94**	**0.94**
Editor	**0.99**	0.96	0.02	**0.62**	0.04	**0.71**
Year	**0.99**	**0.94**	0.98	**0.98**	0.96	**0.99**
Issue	**0.99**	0.98	**0.94**	0.92	**0.96**	0.95
Other	0.89	**0.98**	0.7	**0.93**	**0.8**	0.95
*Place**	N/A	0.95	N/A	0.07	N/A	0.13
macro Average	**0.97**	0.9	0.82	**0.87**	0.83	**0.87**

The above evaluations demonstrate the capability of our method to extract and segment references from a challenging dataset accurately. It indicates also the necessity of training the model on a well-annotated dataset. The details of all evaluations presented in this paper, including datasets, source codes and evaluation metrics, can be found in our public repository[8]. The developed method is employed in the toolchain of EXCITE, which is dedicated to publishing open literature references [158]. In addition, an online demo is made available to the public and can be easily used[13].

3.5 Conclusion

A novel approach for extracting and segmenting references is proposed in this paper. The benefit of combining the different steps in a coherent mechanism is demonstrated and validated with the obtained result. The presented approach is non-parameterized, where it takes the PDF document as input and outputs a list of segmented reference strings. As a result, the approach achieved a satisfactory result on different datasets overcoming state-of-the-art methods. This effectiveness is validated in terms of reference extraction and reference segmentation. Moreover, we introduced a new challenging dataset dedicated to both tasks.

[13] `http://excite.west.uni-koblenz.de/excite`

For future work, on the one side, we will improve our method by combining classical features and word embedding to obtain a better representation of tokens and thus better extraction and segmentation results. On the other side, we will apply the method to a collection of articles, e.g. on German social science, and match them against the records of existing bibliographic databases. The reason is to enrich the citation network with citations, which are unintentionally neglected because of their old publication dates, the linking inability of their publishers, etc.

4 Paper 2, 3 and 4: Metadata Extraction from PDF Documents

MexPub: Deep transfer Learning for Metadata Extraction from German publications

Zeyd Boukhers(✉), *Nada Beili, Timo Hartmann, Prantik Goswami, Muhammad Arslan Zafar*

(DOI: 10.1109/JCDL52503.2021.00076)

Vision and Natural Language for Metadata Extraction from Scientific PDF Documents: a Multimodal Approach

Zeyd Boukhers(✉) *and Azeddine Bouabdallah*

(DOI: 10.1145/3529372.3533295)

TextMap: A Spatial-Semantic Framework for PDF Metadata Extraction and Comparative Performance Analysis

Zeyd Boukhers(✉), *Oya Beyan and Cong Yang*

(DOI: Under Review)

Abstract The availability of metadata for scientific documents is pivotal in propelling scientific knowledge forward and for adhering to the FAIR principles (i.e. Findability, Accessibility, Interoperability, and Reusability) of research findings. However, the lack of sufficient metadata in published documents, particularly those from smaller and mid-sized publishers, hinders their accessibility. This issue is widespread in some disciplines, such as the German Social Sciences, where publications often employ diverse templates. To address this challenge, our study evaluates various feature learning and prediction methods, including natural language processing (NLP), computer vision (CV), and multimodal approaches, for extracting metadata from documents with high template variance. We aim to improve the accessibility of scientific documents and facilitate their wider use. To support our comparison of these methods, we provide comprehensive experimental results, analyzing their accuracy and efficiency in extracting metadata. Additionally, we provide valuable insights into the strengths and weaknesses of various feature learning and prediction methods, which can guide future research in this field.

Keywords: *metadata extraction, document processing, neural networks, natural language processing, computer vision, multimodal approaches, scientific documents*

4.1 Introduction

The widespread availability of scientific metadata has greatly contributed to the success and advancement of the scientific community by enabling the easy findability and accessibility of scientific documents. This is achieved by indexing and linking scientific papers in a large and consistent graph such as the OpenAIRE graph [234] or the Open Research Knowledge Graph [173]. As a result, the field of scientometrics has emerged to study and analyze scholarly literature. While it has become increasingly common for publishers and authors to collect and provide comprehensive metadata alongside the publication of scientific documents, to ensure data's accuracy, completeness, and integrity, this practice was not always popular. Historically, certain disciplines, such as Social Sciences, have seen a considerable portion of their publications become less discoverable due to inadequate metadata collection. This shortfall is particularly evident in works from smaller or mid-sized publishers, which may have lacked the resources or incentive to adequately document metadata, especially in the case of older publications [50, 52]. Consequently, numerous initiatives have been established to consolidate efforts towards enhancing the findability, accessibility, interoperability, and reusability of scholarly data. Prominent among these are The European Open Science Cloud (EOSC)[14], The German Research Data Infrastructure (NFDI)[15] and European Strategy Forum on Research Infrastructures (ESFRI)[16]. The primary focus of these initiatives is on the pivotal task of making metadata universally available.

Alternatively, the metadata can be directly extracted from scientific documents. However, manually extracting metadata from the vast number of published documents is a labour-intensive and time-consuming task, making automation essential. To automate the process, several approaches have been proposed, including classical natural language processing (NLP)-based approaches [355, 155], which aim to extract metadata from PDF documents efficiently and accurately.

With the recent advances in Deep Neural Networks (DNNs) on textual data, significant results have been achieved on this task [254]. This is due to the capability of these networks to capture latent features from the textual documents. However, the problem is still open and far from being solved because scientific documents come in different templates and layouts. This makes it difficult for any model to find common patterns in the order of the

[14] `https://eosc-portal.eu/`
[15] `https://www.nfdi.de/?lang=en`
[16] `https://www.esfri.eu/`

classes. To overcome this problem, some works [50, 13] propose to tackle the problem using image processing techniques and taking advantage of the remarkable advances in computer vision. To this end, these techniques view the scientific PDF documents as RGB images. Furthermore, to harness the strengths of both text and visual information, several studies [52, 29] have adopted multimodal approaches, demonstrating notable effectiveness.

This study explores a variety of feature learning and classification approaches to extract metadata from scientific PDF documents, emphasizing the use of methodologies best suited to the specific challenges of this task. We employ classical approaches such as Conditional Random Fields, advanced NLP techniques including BiLSTM with BERT representations, and innovative multimodal and Textmap methods. While generative LLMs like GPT-4 or LLAMA excel in natural language generation, they are not ideal for structured tasks such as metadata extraction from scientific PDFs. These models, designed primarily for text generation from prompts, face difficulties with fixed formats, which can lead to inaccuracies from over-generalization and context sensitivity, and require substantial resources for task-specific tuning. By contrast, our chosen approaches leverage the strengths of BERT and other architectures to efficiently handle the unique layout variability and multimodal content of scientific documents, ensuring precise and reliable metadata extraction.

In addition to evaluating the technical aspects of these approaches, we also compare their performance and results on a large and unique dataset. One challenge in this area is that many techniques, such as those based on deep neural networks (DNNs), require an extensive ground truth dataset for training. However, creating such a dataset can be difficult, as the process of annotating the data is time-consuming and labor-intensive, and often requires quality checks. To address this issue, we created two challenging datasets, namely, *SSOAR-MVD* and *S-PMRD*. For *SSOAR-MVD*, we synthesized 50.000 samples using a predefined set of templates and available metadata. *S-PMRD* is an authentic subset of the Semantic Scholar Open Research Corpus. The main contributions of this paper are as follows:

- We present a variety of approaches for extracting metadata from scientific PDF documents.
- We created a large, labelled dataset for metadata extraction from scientific PDF documents.
- We conducted extensive experiments to compare the various approaches.
- To facilitate reproducibility and future development, we have made the implementations of all the approaches publicly available[17].

[17] `Willbereleaseduponpublicaiton.`

The remainder of this paper is organized as follows: In Section 4.2, we review related works. In Section 4.3, we introduce all the approaches covered in this paper. Section 4.4 presents the dataset and experimental results, and finally, in Section 4.5, we provide concluding remarks and discuss potential future directions.

4.2 Related Work

Metadata extraction, while a specialized subset of information extraction (IE), serves a distinct purpose and presents unique challenges. This section provides an overview of the most pertinent techniques for metadata extraction, categorizing them into three distinct groups for a clearer understanding of their applications and methodologies.

4.2.1 Natural Language Processing

Metadata extraction in Natural Language Processing (NLP) has primarily been approached through two distinct methodologies: rule-based and machine learning-based techniques [89]. Rule-based techniques rely on predefined rules developed through human expertise to guide metadata extraction [89]. These methods are generally more straightforward to implement but may lack the adaptability found in machine learning-based systems [181, 142]. On the other hand, machine learning-based approaches, exemplified by platforms like CiteSeerX [214], leverage supervised learning algorithms trained on labelled datasets to autonomously extract metadata from new documents. These algorithms range from Hidden Markov Models (HMM) [313], Conditional Random Fields (CRFs) [276], to Support Vector Machines (SVM) [140]. Although robust and effective, the drawback of these machine learning methods lies in the labour-intensive labelling of training data, especially when dealing with samples of high variability.

Recent advancements in Deep Neural Networks (DNN) have provided a new dimension to the field of metadata extraction. DNNs have been shown to considerably outperform traditional methods in effectiveness and efficiency [89]. [165] pioneered a Bidirectional LSTM-CRF model, combining Long Short-Term Memory (LSTM) with a Conditional Random Field (CRF) layer to encode word sequences and predict labels. Similarly, [75] employed a Bidirectional LSTM integrated with a Convolutional Neural Network (CNN) to generate character-level word representations. [17] introduced a DNN-based Segment Sequence Labeling for metadata extraction, setting new performance benchmarks. This approach outstripped existing works such as ParsCit [81], a CRF-based model, and BibPro [70], a neural network-based model, when evaluated on public datasets like UMass [20] and Cora [313].

4.2.2 Computer Vision

While Computer Vision (CV) approaches are not yet ubiquitously applied in the field of metadata extraction, emerging research indicates their promising capabilities, especially for Natural Language Processing (NLP) related tasks. One notable example is DeepPDF [326], which applies a unique perspective to PDF document segmentation. Instead of traditional text-based analysis, DeepPDF treats the document as an image and employs UNet-Zoo, a specialized architecture originally designed for biomedical image segmentation. This approach allows for accurate paragraph identification while ignoring other elements like headers, captions, figures, and references, thus substantiating the potential of CV-based techniques for textual document analysis.

Building upon the groundwork laid by [326], MexPub [49] introduced an innovative technique for extracting metadata from German PDF documents. The methodology utilizes a pixel-by-pixel analysis through the MASK-RCNN architecture [147], specifically engineered for object detection and classification. It incorporates the ResNeXt backbone [370] and Feature Pyramid Networks (FPN) for feature extraction from raw images. While MexPub has shown promising results, it encounters limitations in certain areas. For example, the model struggles with generalizing to scientific literature that diverges structurally from the training dataset. Additionally, MexPub faces challenges in precisely detecting smaller patterns or those placed in unconventional positions. These limitations suggest that the method's performance could be further enhanced by incorporating text processing elements into a unified architecture.

4.2.3 Multimodality

Multimodal deep learning has made significant inroads across various applications, including but not limited to audiovisual and image classification, showcasing impressive performance. Specifically within the realm of metadata extraction, there's growing evidence that multimodal approaches are superior to their unimodal counterparts, as highlighted in studies by [30], [221], and [52].

Balasubramanian et al. [30] employed a combined audio and video modality strategy to extract metadata from video lectures. Their technique harnessed the potential of a Naive Bayes classifier in tandem with a rule-based refiner. The essence of this methodology was capitalizing on the interplay between audio transcripts and the content of slides embedded within video streams. Astonishingly, this synergy yielded a marked 114.2% improvement in metrics such as F-score, precision, and recall when benchmarked against solely audio-based methodologies.

Liu et al. [221] pioneered a multimodal deep-learning strategy tailored for metadata extraction from scientific documents. Their model seamlessly

ingests both image and textual data, negating the need for handcrafted classification features. On the textual front, Recurrent Neural Networks (RNNs) were employed, while image data was processed using Convolutional Neural Networks (CNNs). The amalgamated representation was then processed via a BiLSTM network, culminating in classification through a CRF classifier. The potency of this composite approach was evident when juxtaposed against unimodal strategies.

Further enriching the field,[52] presented an intriguing approach to address metadata extraction challenges specific to German scientific papers, which frequently exhibit a vast array of layouts due to the varied publishing standards of small to mid-sized publishers. The paper proposed a multimodal approach that perceives a PDF document simultaneously as an RGB image and a textual document, using BiLSTM and MexPub, respectively. The outputs from both sub-models are subsequently merged and processed by another BiLSTM model for token classification.

4.3 Approach

This section discusses various feature learning and classification methods for extracting metadata from scientific PDF documents. Like many studies in this area, we assume that metadata may only be present on the first page of a PDF document and that its availability may vary across documents. For example, the Digital Object Identifier (DOI) may not be included in all scientific PDF documents.

Let $\mathcal{P}$ be the first page of a scientific PDF document, consisting of a set of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$, where $n = |\omega|$. Let S be a set of states in a finite state machine, each corresponding to a label $l \in L$ (e.g., *Title*, *Authors*, etc.). The task is to formalize $\gamma(\mathcal{P}) = \mathbf{s}$, where $\mathbf{s} = \langle s_1, s_2, \cdots, s_n \rangle$ is the sequence of states in S that correspond to the labels assigned to the words in the input sequence ω. Table 8 represents the used variables and their descriptions

In this study, we compare several approaches for extracting metadata from scientific PDF documents, including foundational techniques like Conditional Random Fields (CRF) [324] and GROBID [3], which have established the groundwork for metadata extraction. We also implement and explore novel neural sequence labeling approaches using BiLSTM and BiLSTM-CRF architectures (Sections 4.3.2 and 4.3.3). This work introduces three new methodologies that take different approaches to the problem: a computer vision approach using Fast R-CNN (Section 4.3.5), a multimodal neural architecture (Section 4.3.6), and our proposed TextMap framework (Section 4.3.7). All approaches are evaluated following the aforementioned formalization of the metadata extraction task, enabling a comprehensive comparison of their effectiveness.

Variable	Description
γ	The metadata extraction model
$\mathcal{P}$	The first page of the PDF document
$\mathbf{S}$	The outcome of the model, which is a set of strings associated with their labels
y	The metadata label
$\mathbf{s}_i$	The output metadata value of the yth label
K	Section
w	Classified token
ω	Unclassified token

Table 8: Overview of key variables used in this paper across the different feature learning and classification methods for metadata extraction.

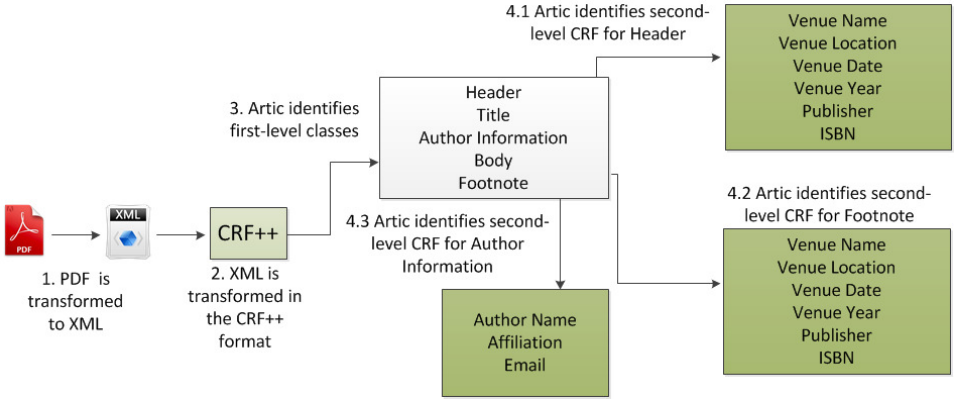


Figure 6: Schematic representation of the two-layer Conditional Random Field (CRF) model for metadata extraction [324].

4.3.1 Conditional Random Fields (CRF)[324]

The approach proposed by Souza, Viviane, and Heuser [324] employs a two-layer Conditional Random Field (CRF) model for extracting metadata from scientific PDF documents. As illustrated in Figure 6, the extraction process is divided into two main steps: identifying main sections and extracting metadata from these sections.

Given the extracted lines from the first page $\mathcal{P}$, the first layer of the CRF model classifies each line into one of the five main sections that may contain metadata information: *Header*, *Title*, *Author Information*, *Body*, and *Footnote*. To achieve this, the model processes font features such as size, style, and alignment from each line and uses them as input. Once the main sections have been identified, the second layer of the CRF model is responsible for extracting metadata from these sections. Some content is automatically excluded from certain sections during this process. For

instance, content that appears in the Footnote section would not be included in the model's output.

For each identified section k, the model processes a sequence of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$. Each word ω_i is represented with a feature vector $\mathbf{x}_i$, which comprises m handcrafted features such as length, whether it follows a year format, presence of special characters, and capitalization, among others. The model calculates the probability of a section sequence given a handcrafted feature sequence using the following equation:

$$P(\mathbf{s} \mid \mathbf{x}) = \frac{\exp\left(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(y_{i-1}, y_i, \mathbf{x}, i)\right)}{Z(\mathbf{x})} \tag{10}$$

where f_j denotes the feature function for the j^{th} feature, and λ_j is the corresponding weight parameter. $Z(\mathbf{x})$ is the normalization factor, ensuring that the sum of probabilities over all possible label sequences equals 1:

$$Z(\mathbf{x}) = \sum_{y'} \exp\left(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(y'_{i-1}, y'_i, \mathbf{x}, i)\right) \tag{11}$$

To find the optimal weights $\lambda_{j=1}^{m}$, a training process is conducted by maximizing the log-likelihood of the training data:

$$\mathcal{L}(\lambda) = \sum_{u=1}^{|D|} \log P\left(y^{(u)}, \mathbf{x}^{(u)}\right) - \frac{\sum_{j=1}^{m} \lambda_j^2}{2\sigma^2} \tag{12}$$

where $(x^{(u)}, y^{(u)})$ are the pair features and label of the u^{th} training instance in the training dataset D, and σ^2 is a hyperparameter for L2 regularization that controls the model's complexity.

4.3.2 Bi-Directional LSTM

For this solution, we employed a Bidirectional Long Short-Term Memory (BiLSTM) model with three layers. The BiLSTM has 112 hidden dimensions and is followed by two fully connected layers. The final layer uses a softmax activation function to assign each word to a specific class. Given a sequence of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$, the embedding vector of each word ω_i is obtained the BERT model:

$$\mathbf{x_i} = \text{BERT}(\omega_i), \quad i = 1, 2, \cdots, n \tag{13}$$

resulting in a sequence of embedding vectors $\mathbf{x} = \langle \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \rangle$.

This sequence of embedding vectors is fed into the three bidirectional LSTM layers with hidden dimensions = 112. Let $\mathbf{h}_i^{(f)}$ and $\mathbf{h}_i^{(b)}$ denote the forward and backward hidden states at position i in the sequence. The hidden states are updated as follows:

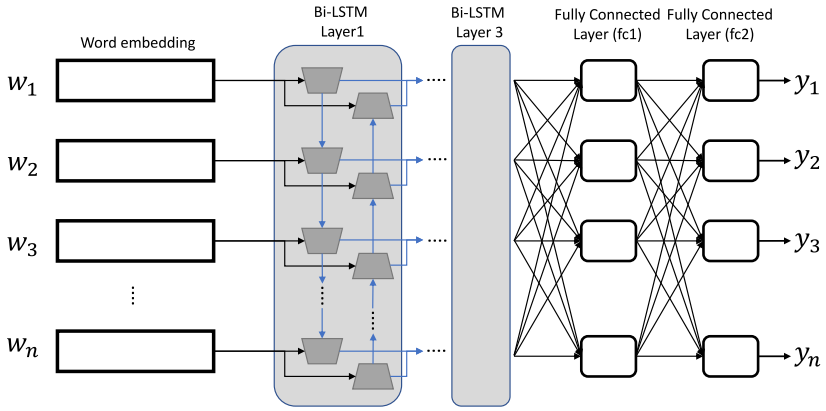


Figure 7: Diagram of the Bi-Directional LSTM network architecture for metadata extraction

$$\begin{aligned} \mathbf{h}_i^{(f)} &= \text{LSTM}^{(f)}(\mathbf{x}_i, \mathbf{h}_{i-1}^{(f)}), \\ \mathbf{h}_i^{(b)} &= \text{LSTM}^{(b)}(\mathbf{x}_i, \mathbf{h}_{i+1}^{(b)}) \end{aligned} \tag{14}$$

For each BiLSTM layer t, the outputs of the forward and backward LSTM units are concatenated to form the hidden state of the BiLSTM layer:

$$\mathbf{h}_i^{(t)} = \left[\mathbf{h}_i^{(f,t)}; \mathbf{h}_i^{(b,t)}\right] \tag{15}$$

The output of the last BiLSTM layer is passed through two fully connected layers with weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ and bias vectors $\mathbf{b}_1$ and $\mathbf{b}_2$:

$$\mathbf{o}_i = \text{ReLU}(\mathbf{W}_2\text{ReLU}(\mathbf{W}_1\mathbf{h}_i^{(\text{BiLSTM})} + \mathbf{b_1}) + \mathbf{b_2}) \tag{16}$$

A softmax activation function is applied to the output of the last fully connected layer to compute the probability distribution over the predefined set of labels for each word in the sequence:

$$\hat{\mathbf{y}}_i = \text{SoftMax}(\mathbf{o}_i) = \frac{\exp(\mathbf{o}_i)}{\sum_{l=1}^{L} \exp(\mathbf{o}_i, l)} \tag{17}$$

4.3.3 BiLSTM-CRF

As BiLSTM-CRF is used in many NLP tasks and specifically extracting information from textual data [15, 83], we assume that it would perform similarly on the task of extracting metadata from PDF documents. Figure 8 illustrates the developed model that takes as input the embeddings of the words extracted from $\mathcal{P}$. The embeddings are obtained using a pre-trained

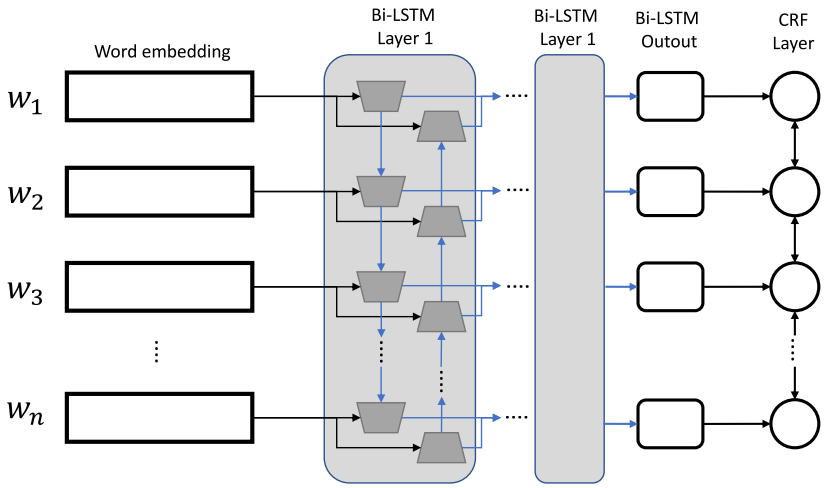


Figure 8: Diagram of the Bi-Directional LSTM network architecture with CRF classifier for metadata extraction

BERT model. The assumption is that most of the metadata classes are represented in structured phrases that BERT is capable to capture. For the other classes (e.g. Author name),

The proposed model consists of a 4-layer BiLSTM network with 115 hidden dimensions followed by a CRF layer for sequence labelling. The sequence of observed words goes through the same steps as mentioned in the Equations 13, 14 and 15. Then, the output of the last BiLSTM layer $H = \sum_i \mathbf{h}_i^{\text{BiLSTM}}$ is passed through a CRF layer to calculate the probability of a label sequence $P(\mathbf{s} \mid H)$, using Equation 10.

4.3.4 Grobid[3]

GROBID is a machine-learning library that is designed to extract, parse, or restructure raw documents into structured XML/TEI documents. It employs a cascade of sequence labelling models to parse each document, allowing it to adapt to the different hierarchical structures present in the documents. By utilizing a cascade approach, GROBID can handle a wide variety of document layouts and structures.

The main idea behind GROBID's approach is to break down the complex task of document parsing into a series of smaller, more manageable tasks. Each model in the cascade focuses on a specific aspect of the document structure, such as headers, titles, author information, or other metadata. The models have a small number of labels, which makes it easier to manage and train. When combined, the full cascade provides a detailed end-result structure.

In GROBID, the models are organized hierarchically to address the inherent hierarchical structure of the documents. The original GROBID model

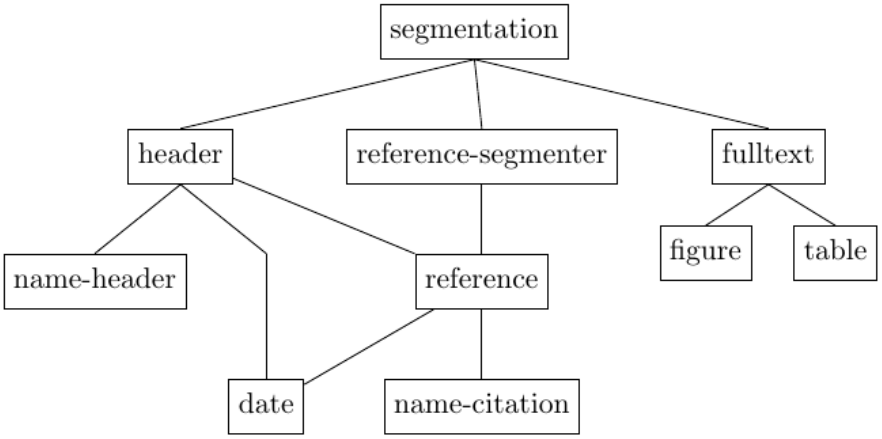


Figure 9: Overview of the GROBID Framework for structured metadata extraction [3]

produces 55 different "leaf" labels, which are the final labels assigned to the text elements after the document has been processed by the entire cascade of models. Each "leaf" label corresponds to a specific element in the structured XML/TEI output.

Model Training: Train a cascade of sequence labeling models on the training dataset. Each model in the cascade is responsible for recognizing and classifying specific elements of the document structure. The models are organized hierarchically, with each model feeding its output to the subsequent model in the cascade.

Model Inference: Given a new document, apply the trained cascade of models to parse the document and extract metadata. The output of each model is fed into the next model in the cascade, refining the document structure at each step. Finally, the "leaf" labels are assigned to the text elements, resulting in a structured XML/TEI representation of the document.

Metadata Extraction: Once the document has been parsed and structured, the metadata can be easily extracted from the XML/TEI representation by querying the relevant elements and their associated "leaf" labels.

4.3.5 Fast-RCNN

In earlier work [50], we addressed this problem by viewing the PDF document as an image and leverage from the advanced progress in computer vision.

The model is an adaptation of Mask R-CNN, a cutting-edge object instance segmentation technique proposed by He et al [146]. It identifies objects within images at the pixel level by extending Faster RCNN with an additional branch for predicting object masks and utilizing Region of Inter-

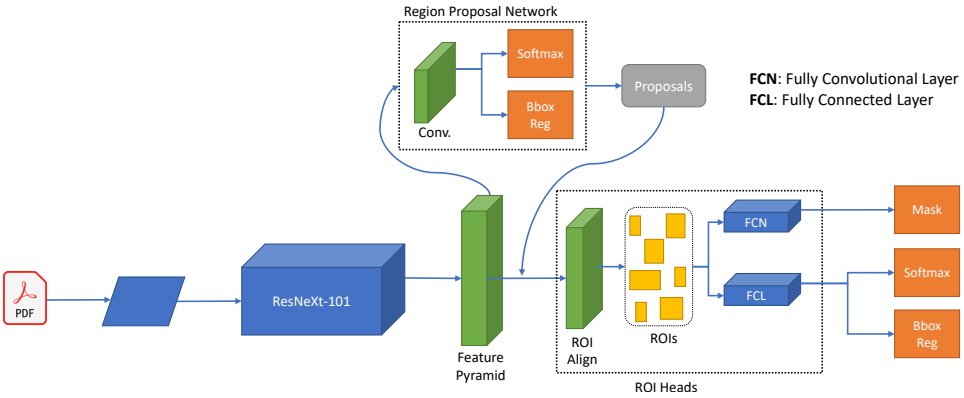


Figure 10: Mask R-CNN architecture employed for metadata extraction from PDF pages.

est (RoI)-Align instead of RoI-Pooling. The binary object mask highlights the position of each object in its bounding box on a pixel-by-pixel basis. In this implementation, Mask R-CNN is combined with a ResNeXt [369] backbone architecture and a Feature Pyramid Network (FPN), following the approach, outlined in [217].

As illustrated in Figure 10, the PDF page $\mathcal{P}$ is first transformed into a pixel image, which serves as input for the RCNN model. The model is composed of three main components: (i) a Feature Pyramid Network (FPN) with ResNeXt as a backbone network, (ii) a Region Proposal Network (RPN), and (iii) RoI (Region of Interest) Heads. As detailed in Table9, the ResNeXt backbone includes a stem block and four stages, each containing multiple bottleneck blocks.

The stem block down-samples the input image twice through a 7×7 convolution with a stride of 2, and max-pooling with a stride of 2, generating a feature map at a 1/4 scale. The subsequent four stages contain bottleneck blocks, each featuring three convolutional layers with kernel sizes of 1×1, 3×3, and 1×1. These stages consist of 3, 4, 23, and 3 bottleneck blocks respectively, and produce feature maps at scales of 1/4, 1/8, 1/16, and 1/32 [369]. A max-pooling layer with a kernel size of 1 and a stride of 2 is introduced to the final stage of ResNeXt, yielding a feature map at a 1/64 scale [146].

The second component, the Region Proposal Network (RPN), suggests candidate object bounding boxes utilizing the outputs from the FPN's five stages. Subsequently, a fully convolutional mask prediction branch is integrated into the head [146]. The RoI head employs fully-connected layers to generate refined box locations and classification results from multiple fixed-size features, which are obtained by cropping and warping feature maps. The box head then filters out up to 100 boxes using non-maximum suppression (NMS) to eliminate redundant detections.

Layer name	scale	kernel size	stride
stem	1/4	7×7	2
backbone 1	1/4	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \\ 1 \times 1 \end{bmatrix} \times 3$	1
backbone 2	1/8	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \\ 1 \times 1 \end{bmatrix} \times 4$	1
backbone 3	1/16	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \\ 1 \times 1 \end{bmatrix} \times 23$	1
backbone 4	1/32	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \\ 1 \times 1 \end{bmatrix} \times 3$	1
max pooling layer	1/64	1×1	2

Table 9: Overview of the ResNeXt structure in the RCNN model, highlighting the number of bottleneck blocks and the scales of feature maps at each stage.

Transfer learning is a widely used technique in deep learning for computer vision tasks. It involves retraining pre-trained convolutional networks on smaller, task-specific datasets to fine-tune the weights and biases, leveraging the knowledge gained from one classification task to another [269]. In our study, we employ a source model based on the Detectron2 [368] implementation of Mask R-CNN ResNeXt-101 32x8d FPN. This model was initially fine-tuned on 191,832 images from the PubLayNet dataset [395], which includes annotated images of articles from PubMed Central™ Open Access (PMCOA) featuring five classes: title, text, list, table, and figure. The model is well-suited for extracting metadata from scientific papers since it (i) has a backbone trained on the extensive COCO dataset, (ii) underwent fine-tuning on a large dataset of scientific document images, and (iii) is designed for a task closely related to ours.

To adapt this model for extracting metadata patterns from scientific documents, we first modified the final layer of the source model to output nine target classes (title, authors, journal, abstract, date, DOI, address, affiliation, and email addresses) instead of the original five. Empirical experiments on a subset of 103 random samples from our training dataset showed that the best-performing architecture has two frozen layers and 15k iterations. Based on these findings, we fine-tuned the model using the full training dataset, setting the learning rate to $2.5 \times 10^{(-3)}$.

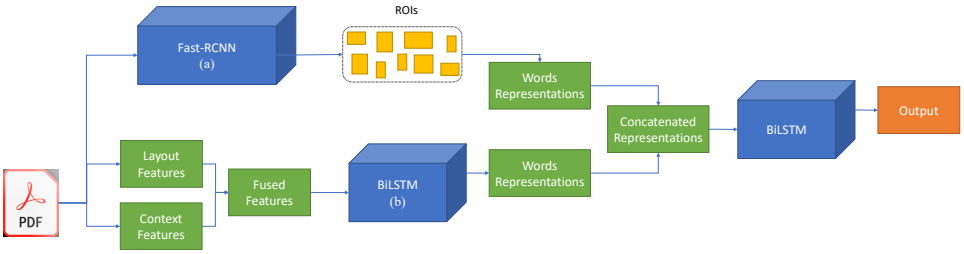


Figure 11: Multimodal extraction approach, where *(a)* refers to the model described in section 4.3.5 and *(b)* refers to the model described in section 4.3.2

4.3.6 Vision and Natural Language

In earlier work [52], we addressed this problem using a multimodal neural network model that employs NLP together with Computer Vision for metadata extraction.

Figure 11 illustrates the initial step of our process, wherein the text is extracted from $\mathcal{P}$ using CERMINE [341]. Known for its reliability in handling diverse layouts at the line level, CERMINE also provides geometric structural information such as text position and font style.

From each extracted token ω, a set of 16 handcrafted features, denoted as F_{hand}, is derived. A word embedding for the token, denoted as F_{embed}, is also generated, which encapsulates the context and meaning of the words. These two sets of features are then concatenated to form a single feature vector, such that $F_{total} = Concat(F_{hand}, F_{embed})$.

The consolidated vector F_{total} is used as the input for the Natural Language Processing (NLP) sub-model, described in section 4.3.2. Simultaneously, the image of $\mathcal{P}$ is supplied as input to the Computer Vision (CV) model, described in section 4.3.5.

Natural Language-based Model To model the extracted text, we utilized a BiDirectional Long-Short-Term Memory (BiLSTM) due to its proven accuracy in handling textual data, as detailed in section 4.3.2. This submodel comprises two layers of LSTM models, each with 256 hidden dimensions; the first layer is a forward LSTM, and the second layer is a backward LSTM. Please refer to section 4.3.2 for more details about BiLSTM

The input to this model is a word representation vector with a length of 1041. As previously described, this vector is the concatenation of two vectors. The first vector, consisting of 16 units, encapsulates layout features such as the font size of the word, font style, the spacing between the word and the line above or below it, and flags denoting whether the text is italicized, bolded, or adheres to a specific common format like date or email, among others. The second vector contains the ELMO [69] embedding results, derived from

a model trained on German documents.

Following the two LSTM layers, a fully connected layer of 512 units is in place, ending with an output layer of 10 neurons, representing the metadata classes. The output layer employs a softmax activation function to generate probability scores for the word's affiliation with each of the classes.

Computer Vision-based Model Building on the proven efficiency of MexPub [50], detailed in section 4.3.5, we leverage it as the Computer Vision (CV) sub-model, feeding it with $\mathcal{P}$. This model yields output in the form of bounding boxes labelled with metadata classes. Subsequently, we extract the text enclosed within the bounding boxes as identified by the CV sub-model and compile the probabilities for all potential classes within that box prior to their submission to the classifier. It's important to note that the CV sub-model may also generate bounding boxes that are unclassified, meaning they do not associate with any of the predefined classes.

Classifier In the final stage of our architecture pipeline, the output of the NLP and CV sub-models is fused using a SoftMax classifier. Specifically, all words from the document are extracted and sequentially traversed. Their vector representations, generated by the sub-models, are then concatenated. A BiLSTM, notable for its bidirectional operation and capability to preserve information from both past and future states, is employed in this context as well. This is especially advantageous for understanding context and discerning patterns within sentences or paragraphs (e.g., if the adjacent words are titles, the current word is highly likely to be a title as well). The model specifically takes in a vector of length 20, resulting from the concatenation of both sub-model outputs. The model's output is a probability distribution of length 10 corresponding to all classes. As depicted in Figure 2, the classifier comprises two stacked LSTM layers (forward and backward LSTMs) each with 256 hidden dimensions. A fully connected layer follows these two layers, encompassing 512 input nodes and 10 output nodes activated by a SoftMax function.

4.3.7 Text Map Approach

The text map approach presents a novel framework that jointly optimizes spatial and semantic information for metadata extraction. Given the first page $\mathcal{P}$ of a PDF document and its sequence of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$, our goal is to learn a mapping function γ that assigns metadata labels while preserving both spatial and semantic relationships.

Two Phase Processing The approach processes documents through two complementary phases as depicted in Figure 12:

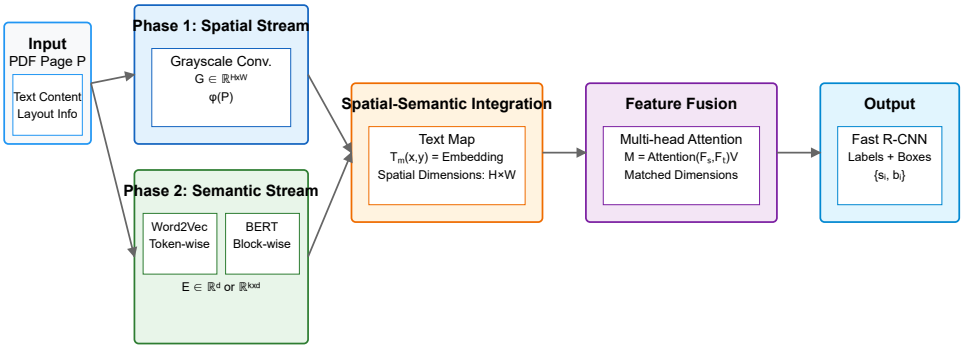


Figure 12: Overview of the TextMap approach

Phase 1: Spatial Representation transforms $\mathcal{P}$ into a grayscale representation G that preserves structural information:

$$G = \phi(\mathcal{P}) \in \mathbb{R}^{H \times W} \tag{18}$$

where H and W are the height and width of the page, respectively. This transformation preserves the spatial distribution of text and structural elements across the document.

Phase 2: Semantic Mapping The semantic mapping differs based on the chosen embedding function ψ. For Word2Vec, each token ω_i is embedded individually:

$$E_i = \psi_{Word2Vec}(\omega_i) \in \mathbb{R}^d \tag{19}$$

For BERT, entire text blocks $B_j = \omega_1, ..., \omega_k j$ are embedded together to capture contextual relationships:

$$E_j = \psi BERT(B_j) \in \mathbb{R}^{k \times d} \tag{20}$$

where d is the embedding dimension, and k is the number of tokens in the block.

Spatial-Semantic Integration The key innovation in our approach is the integration of spatial and semantic information through a carefully designed interpolation process:

1. **Region Identification**: The regions of interest $R = R_1, ..., R_k$ are determined by the locations of text content in the document. Each region R_i corresponds to a bounding box containing embedded text:

$$R_i = \{(x, y, w, h) \mid \text{text content exists at } (x, y) \text{ with width } w \text{ and height } h\} \tag{21}$$

The regions are naturally defined by the presence of text content that has been extracted and embedded. This ensures that our regions directly correspond to actual textual content in the document.

Embedding Interpolation For each region R_i, the embedding is directly mapped into the spatial coordinates of that region to create the text map T_m. We perform a straightforward mapping to ensure that each spatial location in the text map contains the embedding of the text (token or block) that appears at that location in the original document. For Word2Vec, where each token has its own embedding:

$$T_m(x, y) = E_j \quad \text{where} \quad (x, y) \in R_i \text{ contains token } \omega_j \tag{22}$$

For BERT, where entire blocks are embedded together:

$$T_m(x, y) = E_i \quad \text{where} \quad (x, y) \in R_i \text{ contains block } B_i \tag{23}$$

3. **Feature Fusion**: After interpolation, both the grayscale representation G and the text map T_m have matching spatial dimensions, as the embeddings have been mapped to their corresponding regions' coordinates in the document space. Specifically:

$G \in \mathbb{R}^{H \times W}$ from the spatial stream $T_m \in \mathbb{R}^{H \times W \times d}$ from the interpolated embeddings, where d is the embedding dimension

This dimensional alignment allows us to apply convolutional operations to both streams:

$$F_{spatial} = Conv2D(G) \in \mathbb{R}^{H' \times W' \times C} \tag{24}$$

$$F_{semantic} = Conv2D(T_m) \in \mathbb{R}^{H' \times W' \times C} \tag{25}$$

where H' and W' are the reduced spatial dimensions after convolution, and C is the number of output channels. These spatially-aligned feature maps are then fused using a multi-head attention mechanism:

$$M = Attention(F_{spatial}, F_{semantic})V \tag{26}$$

where V is a learnable value matrix. This fusion process effectively combines the structural information from the spatial stream with the semantic information from the text embeddings, while maintaining spatial correspondence between the two streams.

Segmentation and Classification The fused features M are processed through a Fast R-CNN architecture for final segmentation and classification. For each identified region, we predict both the class label and bounding box refinements:

$$\{s_i, b_i\} = FastRCNN(R_{refined}) \tag{27}$$

where s_i is the metadata label and b_i are the refined coordinates.

Joint Optimization Framework The model is trained through a joint optimization framework that combines three objectives:

1. **Semantic Objective** ($\mathcal{L}_{semantic}$):

$$\mathcal{L}_{semantic}(\theta) = -\sum_{i=1}^{n} \log P(s_i|\omega_i, E_i) \tag{28}$$

This term ensures accurate metadata label assignment based on textual content.

2. **Spatial Objective** ($\mathcal{L}_{spatial}$):

$$\mathcal{L}_{spatial}(\theta) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} \|f(G_i) - f(G_j)\|_2^2 \cdot \mathbb{1}[s_i = s_j] \tag{29}$$

where $\mathcal{N}(i)$ represents the spatial neighbors of token i, f is a feature extraction function, and $\mathbb{1}$ is the indicator function.

3. **Cross-modal Objective** ($\mathcal{L}_{cross}$):

$$\mathcal{L}_{cross}(\theta) = -\sum_{i=1}^{n} \log P(s_i|E_i, f(G_i)) \tag{30}$$

The complete optimization objective is:

$$\mathcal{J}(\theta) = \alpha\mathcal{L}_{semantic}(\theta) + \beta\mathcal{L}_{spatial}(\theta) + \gamma\mathcal{L}_{cross}(\theta) \tag{31}$$

where α, β, and γ are learnable parameters that balance the contribution of each term.

Training and Inference During training, we optimize $\mathcal{J}(\theta)$ using mini-batch stochastic gradient descent:

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta\mathcal{J}(\theta_t) \tag{32}$$

where η is the learning rate.

At inference time, for a new document page $\mathcal{P}$, we: 1. Generate spatial features $G = \phi(\mathcal{P})$ 2. Compute embeddings $E_i = \psi(\omega_i)$ for each token 3. Identify regions 4. Apply the trained model to obtain metadata labels:

$$s_i = \arg\max_{l \in L} P(l|E_i, f(G_i)) \tag{33}$$

This formulation ensures that:

- Tokens with similar semantics and spatial proximity are likely to share labels
- The model can handle variable document layouts
- Both local and global document structure are considered
- The extraction is robust to variations in formatting

4.4 Experiments

In this section, we compare the performance of the described methods in the previous section on two challenging datasets.

4.4.1 Dataset

This section presents a comparative analysis of the different methodologies aimed at extracting metadata from academic PDF documents. To this end, we prepared two challenging datasets, namely SSOAR-MVD and S-PMRD.

SSOAR Multidisciplinary Vision Dataset (SSOAR-MVD) To ensure a fair comparison of all the methods described in the section, we ensured they were all applied to the same dataset. As a result, we collected a challenging dataset of 50,000 documents from the SSOAR repositor[18]. The SSOAR stores publications from various publishers, including small and mid-sized ones, covering a range of disciplines known for their challenging layout formats, such as Social Sciences, Humanities, Law, and Administration. This guarantees that a wide variety of templates are included in the dataset. During the scraping process, each document was downloaded along with its textual metadata provided by the SSOAR repository. However, since most computer vision approaches require labelled images (i.e., bounding boxes), a preprocessing phase was conducted to ensure this. Each document underwent the following steps:

- The document is converted into an image.
- Using an open-source tool provided by TensorFlow, blocks of text were extracted from the document along with their respective bounding boxes.
- The similarity between each text block and metadata class was measured (using the collected textual metadata from the SSOAR).
- If a certain block had a near-perfect similarity with a specific class, the corresponding bounding box was assigned to that class. Otherwise, it was assigned a class "other".

S2ORC PDF Metadata Refinement Dataset (S-PMRD) To evaluate the efficacy of these methods on an authentic corpus, we meticulously curated a subset from the Semantic Scholar Open Research Corpus (S2ORC) [226]. While S2ORC offers a vast repository of millions of scholarly articles, our preliminary analyses revealed significant discrepancies between the raw textual data provided by S2ORC and the content within the corresponding

[18] `https://www.gesis.org/en/ssoar/home`

PDF documents. Notably, certain text segments available in the S2ORC dataset were absent from the PDFs, and this does not explain the extraction methodologies employed by S2ORC. These inconsistencies pose substantial challenges for conducting detailed academic analyses that necessitate precise text alignment, such as citation context analysis, text-based data mining, and metadata extraction.

To address these challenges, we developed a specialized sub-corpus of S2ORC, specifically aimed at extracting metadata directly from PDF documents, thereby bypassing the potential inaccuracies inherent in pre-extracted text. The processing pipeline for each document included in this sub-corpus is as follows:

- Using the Digital Object Identifier (DOI) from S2ORC, additional metadata, including links to the actual PDF documents, is retrieved from CrossRef [149] using their API.
- PDFs are downloaded when available, acknowledging that some links may be inactive or access-restricted.
- Text is extracted from the first page of each PDF. This extracted text undergoes a normalization process to eliminate irregularities such as inconsistent spacing and line breaks, thereby ensuring uniformity across the dataset.
- We employ both exact and fuzzy matching techniques to extract critical metadata elements, including author names, titles, abstracts, and affiliations. This dual-method approach accommodates minor discrepancies due to text recognition errors or formatting variations.
- Each document is converted into an image representation.
- For each identified metadata element, bounding boxes are determined based on their locations within the text.

Ultimately, each instance in this dataset comprises:

- The original PDF file.
- The normalized text extracted from the PDF.
- An image of the first page of the PDF.
- Metadata attributes annotated with both their positions in the extracted text and their coordinates on the corresponding image.

4.4.2 Settings

In this study, we employ a token-level evaluation to validate each model's effectiveness, where each predicted token is compared against the ground truth annotations in our dataset. This evaluation is conducted using standard metrics such as Precision, Recall, and F1-Score.

4.4.3 Results

In the first analysis, we evaluate the performance of the presented methods on the datasets SSOAR-MVD and S-PMRD, as depicted in Tables Tables 10 and 11, illustrating the outcomes in terms of Precision, Recall, and F1-Score to provide a multi-dimensional comparison. Notably, the proposed TextMap-Word2Vec method achieves the highest F1-score of 0.913, suggesting that capturing both the semantics of the PDF content and its layout is particularly effective for this task. This is evident by the performance of the Fast-RCNN and Vision-Langauge methods which also leverage the layout and the appearance of the PDF document. In contrast, traditional models like CRF exhibit lower performance metrics, which may indicate difficulties in adapting to the multi-faceted nature of PDF content, where layout and semantic context play crucial roles. An important observation is that the embedding approach in TextMap plays a significant role in the performance of the model.

To give a better overview of the performance of each method, we present below the result of each method for each attribute. Table 12 presents the detailed results of CRF on SSOAR-MVD. As demonstrated, CRF excels in attributes with structured and predictable formats such as Dates (F1-score: 0.731) and DOIs (F1-score: 0.749), where the patterned nature of the data plays to the strengths of the CRF's sequence modelling capabilities. However, CRF struggles with more complex and less structured text such as Titles (F1-score: 0.433) and Abstracts (F1-score: 0.392), where the variability in content and formatting challenges its ability to accurately predict boundaries and content. The moderate success in extracting Authors (F1-score: 0.482) and higher performance in Affiliation data (F1-score: 0.672) suggest that CRF can handle semi-structured text effectively when patterns are somewhat predictable.

Table 13 presents the detailed results of Bi-LSTM on SSOAR-MVD. The BiLSTM method exhibits strong performance across several metadata categories such as 'Title', 'Author', etc. This reflects its robust capability to capture both the context and sequence of text within scholarly PDF documents. Specifically, the method maintains strong performance in handling Abstracts and Dates, with F1-scores slightly above 0.910. This indicates BiLSTM's adeptness at managing narrative content and specific formatted text. However, the slightly lower performance in extracting Affiliation data,

Table 10: Overall Performance Comparison of Different Methods on SSOAR-MVD

Method	**Precision Macro**	**Precision Micro**	**Recall Macro**	**Recall Micro**	**F1-score**
CRF	0.609	0.544	0.524	0.471	0.57
BiLSTM	0.901	0.89	0.898	0.861	0.9
LSTM-CRF	0.778	0.713	0.745	0.697	0.761
GROBID	0.854	0.671	0.794	0.551	0.821
Fast-RCNN	0.9	0.915	0.896	0.904	0.898
Vision-Language	0.935	**0.94**	0.902	0.904	0.92
TextMap-Bert	0.908	0.887	0.902	0.897	0.905
TextMap-Word2Vec	**0.917**	0.92	**0.91**	**0.904**	**0.913**
TextMap-Char2Vec	0.845	0.8	0.849	0.797	0.847

Table 11: Overall Performance Comparison of Different Methods on S-PMRD

Method	**Precision Macro**	**Precision Micro**	**Recall Macro**	**Recall Micro**	**F1-score**
CRF	0.573	0.521	0.501	0.45	0.511
BiLSTM	0.883	0.872	0.898	0.863	0.889
LSTM-CRF	0.740	0.707	0.736	0.692	0.724
GROBID	0.822	0.651	0.787	0.542	0.791
Fast-RCNN	0.874	0.906	0.886	0.912	0.893
Vision-Language	0.91	**0.923**	0.904	0.911	0.903
TextMap-Bert	0.882	0.860	0.916	0.887	0.894
TextMap-Word2Vec	**0.892**	0.902	**0.91**	**0.902**	**0.901**
TextMap-Char2Vec	0.815	0.786	0.841	0.792	0.821

Table 12: Performance Metrics for CRF Method on SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.568	0.35	0.433
Abstract	0.457	0.344	0.392
Authors	0.57	0.418	0.482
Email	0.612	0.607	0.609
Address	0.522	0.481	0.5
Date	0.754	0.71	0.731
Journal	0.547	0.577	0.561
Affiliation	0.663	0.682	0.672
DOI	0.795	0.709	0.749
Macro Average	0.609	0.524	0.57
Micro Average	0.544	0.471	N/A

with an F1-score of 0.833, suggests some challenges in dealing with categories of short strings.

Table 13: Performance Metrics for BiLSTM Method on SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.931	0.91	0.920
Abstract	0.908	0.914	0.911
Authors	0.944	0.93	0.937
Email	0.881	0.86	0.870
Address	0.9	0.882	0.891
Date	0.916	0.905	0.910
Journal	0.865	0.891	0.878
Affiliation	0.814	0.853	0.833
DOI	0.952	0.94	0.946
Macro Average	0.901	0.898	0.9
Micro Average	0.89	0.861	N/A

Table 14 presents the detailed results of LSTM-CRF on SSOAR-MVD. The LSTM-CRF method demonstrates moderate efficacy in extracting different metadata categories, notably outperforming traditional CRF models that rely on handcrafted features. This enhancement suggests that the LSTM architecture provides a more robust feature representation for CRF to utilize effectively. However, despite its competencies across various categories, LSTM-CRF does not surpass the BiLSTM method, which exhibits superior handling of long-term sequential and contextual data dependencies.

Table 15 presents the detailed results of Grobid on SSOAR-MVD. Despite its simple design, GROBID exhibits robust performance across different categories. It particularly excels in extracting Authors and Abstracts, achieving

Table 14: Performance Metrics for LSTM-CRF Method on SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.741	0.699	0.719
Abstract	0.688	0.7	0.693
Authors	0.84	0.815	0.827
Email	0.801	0.782	0.791
Address	0.725	0.76	0.742
Date	0.89	0.822	0.854
Journal	0.739	0.727	0.732
Affiliation	0.774	0.68	0.723
DOI	0.81	0.724	0.764
Macro Average	0.778	0.745	0.761
Micro Average	0.713	0.697	N/A

impressive F1 scores of 0.958 and 0.935 respectively. These high scores can be attributed to GROBID's effective application of its cascading sequence labelling models, which are adept at handling well-structured and clearly delineated data. However, GROBID encounters some variability in categories involving more complex or less standardized information, such as Address and Affiliation. This variability stems from these categories' inherent challenges, including inconsistent formatting and multifaceted data structures that can complicate the parsing process. While the cascading approach of GROBID generally enhances its capability to manage hierarchical document structures efficiently, it occasionally struggles with elements that lack a clear or uniform presentation.

Table 15: Performance Metrics for GROBID Method on SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.764	0.667	0.951
Abstract	0.84	0.79	0.935
Authors	0.934	0.855	0.958
Email	0.91	0.812	0.893
Address	0.722	0.78	0.872
Date	0.855	0.877	0.873
Journal	0.887	0.75	0.927
Affiliation	0.859	0.813	0.818
DOI	0.911	0.8	0.916
Macro Average	0.854	0.794	0.821
Micro Average	0.671	0.551	N/A

Table 16 presents the detailed results of Fast-RCNN on SSOAR-MVD. The model demonstrates high precision and recall across most categories,

with solid performance in the Title, Abstract, and Journal categories, suggesting robustness in recognizing well-defined metadata fields. The Email and Authors categories also show commendable accuracy. However, the model indicates slightly weaker performance in the Address and Date categories, with F1-scores of 0.837 and 0.832, respectively. This could be due to variability in the formatting and presentation of these metadata elements across documents, which poses challenges in consistent extraction. The small variance between Macro Average and Micro Average metrics indicates a balanced performance across different categories, without significant bias toward any particular type of metadata.

Table 16: Performance Metrics for Fast-RCNN Method on SSOAR-MVD

Category	**Precision Macro**	**Recall Macro**	**F1-score**
Title	0.966	0.95	0.958
Abstract	0.915	0.922	0.918
Authors	0.91	0.938	0.924
Email	0.933	0.917	0.925
Address	0.875	0.802	0.837
Date	0.825	0.84	0.832
Journal	0.94	0.925	0.932
Affiliation	0.839	0.876	0.857
DOI	0.901	0.893	0.897
Macro Average	0.9	0.896	0.898
Micro Average	0.915	0.904	N/A

Tables 17, 18, 19 present the performance metrics across three different configurations of the TextMap model, using BERT, Word2Vec, and Char2Vec embeddings. we can observe varied performances that highlight the strengths and weaknesses of each embedding strategy with the TextMap approach. TextMap using BERT Embeddings configuration (Table 17) demonstrates strong performance across several categories, particularly in Authors, Abstract, and Journal, with F1-scores > 0.92. This suggests that BERT's deep contextual embeddings are particularly effective at extracting structured text like titles and authors' details, typically well-defined in the document. The lower performance in the Affiliation and Address categories, with relatively lower F1 scores, could indicate challenges in capturing less consistently formatted information.

TextMap using Word2Vec Embeddings configuration (Table 18) generally performs well, particularly in the Authors and Abstract categories. This suggests good generalization in capturing both the semantic and structural patterns in data, though slightly less effectively than BERT in terms of overall averages. However, as shown in Table 20, it has a lower computational cost in both training and inference. Consequently, this configuration provides

a good balance between performance and computational efficiency, especially suitable for environments where computational resources or training data are limited.

TextMap using Char2Vec Embeddings configuration (Table 19) demonstrates a notable decline in performance across most categories compared to the BERT and Word2Vec models. It performs best in the DOI category with an F1-score > 0.9 but struggles particularly with Abstract and Journal metadata, with F1-scores < 0.8. In conclusion, Char2Vec, while useful in certain niche applications (like OCR and typo-sensitive extractions), may not be suitable for tasks requiring deep semantic understanding such as understanding the semantics of a scholarly text.

Table 17: Performance Metrics for TextMap using Bert embeddings.

Category	**Precision Macro**	**Recall Macro**	**F1-score**
Title	0.954	0.949	0.951
Abstract	0.921	0.95	0.935
Authors	0.967	0.949	0.958
Email	0.91	0.877	0.893
Address	0.889	0.856	0.872
Date	0.855	0.891	0.873
Journal	0.924	0.93	0.927
Affiliation	0.822	0.815	0.818
DOI	0.931	0.902	0.916
Macro Average	0.908	0.902	0.905
Micro Average	0.887	0.897	N/A

Table 18: Performance Metrics for TextMap using Word2Vec embeddings

Category	**Precision Macro**	**Recall Macro**	**F1-score**
Title	0.962	0.922	0.941
Abstract	0.933	0.952	0.942
Authors	0.978	0.949	0.963
Email	0.93	0.899	0.914
Address	0.904	0.86	0.881
Date	0.852	0.907	0.878
Journal	0.924	0.94	0.931
Affiliation	0.834	0.849	0.841
DOI	0.933	0.915	0.923
Macro Avrerage	0.917	0.91	0.913
Micro Average	0.92	0.904	N/A

In addition to comparing the models' performance in terms of precision,

Table 19: Performance Metrics for TextMap using Char2Vec embeddings

Category	**Precision Macro**	**Recall Macro**	**F1-score**
Title	0.851	0.874	0.862
Abstract	0.77	0.8	0.785
Authors	0.849	0.815	0.832
Email	0.902	0.89	0.896
Address	0.86	0.881	0.870
Date	0.875	0.842	0.858
Journal	0.782	0.809	0.795
Affiliation	0.804	0.83	0.817
DOI	0.917	0.905	0.911
Macro Average	0.845	0.849	0.847
Micro Average	0.8	0.797	N/A

recall, and F1 score, we compare their computational complexity using the SSOAR-MVD dataset.

This comparison reveals a range of trade-offs between computational efficiency and performance accuracy. CRF offers the quickest inference time and requires the least training time, making them ideal for environments where speed is prioritized over cutting-edge accuracy. BiLSTM models, while requiring more extensive training, provide rapid inference capabilities, suitable for real-time applications once the model is deployed. LSTM-CRF models combine the deep learning prowess of LSTMs with the structured output of CRFs did not achieve higher accuracy compared to Bi-LSTM models and has longer training times and moderately slow inference speeds. GROBID, tailored specifically for document processing tasks, demands the most extended training period and exhibits slower inference times, reflecting its comprehensive analytical depth. Fast R-CNN, effective in precise localization of content within documents, also shows a moderate training duration with slower inference, suited to applications where precision is more critical than speed. Vision-language models, though offering superior performance where an understanding of both visual cues and textual information is necessary, involve the longest training durations and the slowest inference rates, which could be a significant drawback in time-sensitive scenarios.

Lastly, TextMap models using BERT, Word2Vec, and Char2Vec embeddings demonstrate a spectrum of efficiencies, with BERT providing high accuracy but slower inference and longer training times, whereas Word2Vec offers a more balanced approach, making it preferable for scenarios that demand both efficiency and effectiveness.

Table 20: Average Training and Inference Times for Different Machine Learning Models Used in Metadata Extraction. The table lists the estimated average training and inference times for each model and standard deviations for these estimates.

Model	Training time	Inference time
CRF	36 ± 7.2 hours	0.5 ± 0.01 seconds
BiLSTM	84 ± 7.1 hours	0.1 seconds
LSTM-CRF	126 ± 3.7 hours	0.2 seconds
GROBID	156 ± 7.2 hours	1.2 ± 0.6 seconds
Fast-RCNN	60 ± 6.8 hours	1.3 ± 0.4 seconds
Vision-Language	192 ± 14.5 hours	3.5 ± 1.11 seconds
TextMap-Bert	172 ± 13.3 hours	1.3 ± 0.58
TextMap-Word2Vec	92 ± 6.2	0.4 seconds
TextMap-Char2Vec	90 ± 7.0	0.4 seconds

4.4.4 Limitations

While the models examined in this study demonstrate considerable potential for metadata extraction from scholarly PDF documents, several limitations must be acknowledged to fully appreciate their applicability and scope of use.

- **Dependency on Training Data**: All models, particularly deep learning-based ones like BiLSTM, LSTM-CRF, and TextMap with BERT embeddings, exhibit a high dependency on the quantity and quality of the training data. Their performance is contingent upon the availability of large, annotated datasets. This reliance can limit their practical deployment in scenarios where such datasets are not readily available or are too domain-specific.

- **Adaptability to Rapid Changes**: The field of digital publishing is evolving, with new standards and formats emerging. The adaptability of these models to such rapid changes has not been thoroughly tested, raising concerns about their long-term viability without continuous updates and retraining.

  Error Propagation: In multi-stage models like GROBID or Fast-RCNN, errors in early processing stages can propagate, leading to compounded errors in metadata extraction outcomes. This cascade effect can significantly affect the overall quality of the extracted metadata.

4.5 Conclusion

This study has conducted a comprehensive comparison of various machine learning models to evaluate their effectiveness in extracting metadata from two challenging datasets. The analysis revealed significant variations in performance and computational demands across the models, underscoring the importance of selecting an appropriate model and architecture tailored to specific use case requirements.

The CRF and BiLSTM models demonstrated rapid inference capabilities coupled with robust performance, making them ideal candidates for real-time applications. In contrast, the LSTM-CRF hybrid model, despite combining the strengths of LSTMs and the structured output capabilities of CRFs, did not achieve results on par with its component technologies.

Models that integrate vision and language modalities, while resource-intensive, deliver depth and precision in analysis that simpler models cannot achieve. This sophistication makes them particularly valuable in scenarios where the accuracy of extracted metadata critically impacts the outcomes of subsequent processes.

The TextMap models, which leverage various embeddings such as BERT, Word2Vec, and Char2Vec, offer a spectrum of choices balancing training and inference times with performance. Among these, BERT embeddings stand out for their exceptional accuracy, albeit at a higher computational cost, illustrating the fundamental trade-offs between resource investment and extraction efficacy.

Ultimately, the selection of a metadata extraction model should be driven not only by dataset characteristics but also by the practical constraints of the use case—available computational resources, required inference speed, and the trade-offs between precision and performance that stakeholders are prepared to accept. Future research should consider the potential of hybrid models and the development of more efficient training algorithms to further optimize the application of machine learning in metadata extraction tasks, enhancing both their efficiency and accessibility.

5 Papers 3 and 4: Author Name Disambiguation

Whois? Deep Author Name Disambiguation using Bibliographic Data

Zeyd Boukhers(✉) *and Nagaraj Bahubali Asundi*

(DOI: 10.1007/978-3-031-16802-4 _16) -Best Paper Award-

Deep Author Name Disambiguation using DBLP Data

Zeyd Boukhers(✉) *and Nagaraj Bahubali Asundi*

(DOI: 10.1007/s00799-023-00361-6)

Abstract In the academic world, the number of scientists grows every year and so does the number of authors sharing the same names. Consequently, it challenging to assign newly published papers to their respective authors. Therefore, Author Name Ambiguity (ANA) is considered a critical open problem in digital libraries. This paper proposes an Author Name Disambiguation (AND) approach that links author names to their real-world entities by leveraging their co-authors and domain of research. To this end, we use data collected from the DBLP repository that contains more than 5 million bibliographic records authored by around 2.6 million co-authors. Our approach first groups authors who share the same last names and same first name initials. The author within each group is identified by capturing the relation with his/her co-authors and area of research, represented by the titles of the validated publications of the corresponding author. To this end, we train a neural network model that learns from the representations of the co-authors and titles. We validated the effectiveness of our approach by conducting extensive experiments on a large dataset.

Keywords: *author name disambiguation, entity linkage, bibliographic data, neural networks, classification, DBLP*

5.1 Introduction

AND is an important task in digital libraries that aims to properly link each publication to its respective co-authors so that author-level metrics can be accurately calculated and authors' publications can be easily found. However, this task is highly challenging due to the high number of authors sharing the same names. In this paper, *author name* denotes a sequence of characters referring to one or several authors [19], whereas *author* refers to a unique

[19]It is estimated that about 114 million people share 300 common names.

person authoring at least one publication and cannot be identified only by his/her *author name* [20] but rather with the support of other identifiers such as ORCID, ResearchGate ID and Semantic Scholar author ID.

Although relying on these identifiers almost eliminates any chance of mislinking a publication to its appropriate author, most bibliographic sources do not include such identifiers. This is because not all of the authors are keen to use these identifiers and if they are, there is no procedure or policy to include their identifiers when they are cited. Therefore, in bibliographic data (e.g. references), authors are commonly referred to by their names only. Considering the high number of authors sharing the same names (i.e. homonymy), it is difficult to link the names in bibliographic sources to their real-world authors especially when the source of the reference is not available or does not provide indicators of the author's identity. The problem is more critical when names are substituted by their initials to save space, and when they are erroneous due to wrong manual editing. Disciplines like social sciences and humanities suffer more from this problem as most of the publishers are small and mid-sized and cannot ensure the continuous integrity of the bibliographic data.

Table 21 demonstrates real examples of reference strings covering the problems mentioned above. The homonomy issue shows an example of two different papers citing the name *J M Lee* which refers to two different authors. In this case, it is not possible to disambiguate the two authors without leveraging other features. The Synonymy issue shows an example of the same author *Jang Myung Lee* cited differently in two different papers as *Jang Myung Lee* and *J Lee*. Synonymy is a serious issue in the author name disambiguation as it requires the awareness of all name variates of the given author. Moreover, some name variates might be shared by other authors, which increases homonymy.

Since these problems are known for decades, several studies [248, 188, 116, 168, 113, 289, 385, 182, 183] have been conducted using different machine learning approaches. This problem is often tackled using supervised approaches such as Support Vector Machine (SVM) [141], Bayesian Classifi-

[20] In the DBLP database, there are 27 exact matches of 'Chen Li', 23 reverse matches and more than 1000 partial matches

[21] Xu, Zhihao, et al. "Teleoperating a formation of car-like rovers under time delays." Proceedings of the 30th Chinese Control Conference. IEEE, 2011.

[22] Shi, Pu, Jianning Hua, and Yiwen Zhao. "Posture-based virtual force feedback control for teleoperated manipulator system." 2010 8th World Congress on Intelligent Control and Automation. IEEE, 2010.

[23] Xu, Zhihao, Lei Ma, and Klaus Schilling. "Passive bilateral teleoperation of a car-like mobile robot." 2009 17th Mediterranean Conference on Control and Automation. IEEE, 2009.

[24] Lu, Ching-Hsi, Hong-Yang Hsu, and Lei Wang. "A new contrast enhancement technique by adaptively increasing the value of histogram." 2009 IEEE international workshop on imaging systems and techniques. IEEE, 2009.

Table 21: Illustrative examples of author name ambiguity and incorrect author names

Issue Type	Source	Citations
Synonyms	See [21]	T. Jin, **J. Lee**, and H. Hashimoto, "Internet-based obstacle avoidance of mobile robot using a force-reflection," in Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, (Sendai, Japan), pp. 3418– 3423, October 2004.
	See [22]	TasSeok Jin, **JangMyung Lee**, and Hideki Hashimoto, "Internet-based obstacle avoidance of mobile robot using a force-reflection," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3418-3423. 2004.
Homonyms	See [23]	T.S. Jin, **J.M. Lee**, and H. Hashimoto. Internet-based obstacle avoidance of mobile robot using a force-reflection. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3418–3423, Sendai, Japan, October 2004.
	See [24]	H-J Kim, **J-M Lee**, J-A Lee, S-G Oh, W-Y Kim, "Contrast Enhancement Using Adaptively Modified Histogram Equalization", Lecture Notes in Computer Science, Vol.4319, pp.1150 - 1158, Dec. 2006.

cation [385] and Neural networks (NN) [345]. These approaches rely on the matching between publications and authors which are verified either manually or automatically. Unsupervised approaches [222, 187, 109] have also been used to assess the similarity between a pair of papers. Other unsupervised approaches are also used to estimate the number of co-authors sharing the same name [392] and decide whether new records can be assigned to an existing author or a new one [289]. Due to the continuous increase of publications, each of which cites tens of other publications and the difficulty to label this streaming data, semi-supervised approaches [228, 393] were also employed. Recent approaches [384, 372] leveraged the outstanding efficiency of deep learning on different domains to exploit the relationship among publications using network embedding. All these approaches use the available publication data about authors such as titles, venues, year of publication and affiliation. Some of these approaches are currently integrated into different bibliographic systems. However, all of them require an exhausting manual correction to reach an acceptable accuracy. In addition, most of these ap-

proaches rely on the metadata extracted from the papers which are supposed to be correct and complete. In real scenarios, the source of the paper is not always easy to find and only the reference is available.

In this paper, which builds upon our earlier work [47], we aim to employ bibliographic data consisting of publication records to link each author's name in unseen records to their appropriate real-world authors (i.e. DBLP identifiers) by leveraging their co-authors and area of research embedded in the publication title and source. Note that the goal of this paper is to disambiguate author names in newly published papers that are not recorded in any bibliographic database. Therefore, all records that are considered unseen are discarded from the bibliographic data and used only for testing the approach. The assumption is that any author is most likely to publish articles in specific fields of research. Therefore, we employ articles' titles and sources (i.e. Journal, Booktitle, etc.) to bring authors close to their fields of research represented by the titles and sources of publications. We also assume that authors who already published together are more likely to continue collaborating and publishing other papers.

For the goal mentioned above, our proposed model *WhoIs* is trained on a bibliographic collection obtained from DBLP, where a sample consists of a target author, pair of co-authors, title and source. For co-authors, the input is a vector representation obtained by applying Char2Vec which returns character-level embedding of words. For title and source, the BERT model is used to capture the semantic representations of the sequence of words. Our model is trained and tested on a challenging dataset, where thousands of authors share the same atomic name variate. The main contributions of this paper are:

- We proposed a novel approach for author name disambiguation using semantic and symbolic representations of titles, sources, and co-authors.
- We provided a statistical overview of the problem of author name ambiguity.
- We conducted experiments on challenging datasets simulating a critical scenario.
- The obtained results and the comparison against baseline approaches demonstrate the effectiveness of our model in disambiguating author names.

The rest of the paper is organized as follows. Section 5.2 briefly presents related work. Section 5.3 describes the proposed framework. Section 5.4 presents the dataset, implementation details and the obtained results of the proposed model. Finally, Section 5.5 concludes the paper and gives insights into future work.

5.2 Related Work

In this section, we discuss recent approaches softly categorized into three categories, namely unsupervised-, supervised- and graph-based;

5.2.1 Unsupervised-based:

Most of the studies treat the problem of author name ambiguity as an unsupervised task [187, 392, 183, 183, 289] using algorithms like DBSCAN [183] and agglomeration clustering [363]. Liu et al. [222] and Kim et al. [187] rely on the similarity between a pair of records with the same name to disambiguate author names on the PubMed dataset. Zhang et al. [392] used Recurrent Neural Network (RNN) to estimate the number of unique authors in the Aminer dataset. This process is followed by manual annotation. In this direction, Ferreira et al. [115] have proposed a two-phase approach applied to the DBLP dataset, where the first one is obtaining clusters of authorship records and then disambiguation is applied to each cluster. Wu et al. [363] fused features such as affiliation and content of papers using Shannon's entropy to obtain a matrix representing pairwise correlations of papers which is in return used by Hierarchical Agglomerative Clustering (HAC) to disambiguate author names on Arnetminer dataset. Similar features have been employed by other approaches [376, 22].

5.2.2 Supervised-based:

Supervised approaches [141, 288, 331, 345, 385] are also widely used but mainly only after applying to block that gathers authors sharing the same names together. Han et al. [141] present two supervised learning approaches to disambiguate authors in cited references. Given a reference, the first approach uses the Naive Bayes model to find the author class with the maximal posterior probability of being the author of the cited reference. The second approach uses SVM to classify references from DBLP to their appropriate authors. Sun et al. [331] employ heuristic features like the percentage of citations gathered by the top name variations for an author to disambiguate common author names. Neural networks are also used [345] to verify if two references are close enough to be authored by the same target author or not. Hourrane et al. [159] propose a corpus-based approach that uses word embeddings to compute the similarity between cited references. In [103], an Entity Resolution system called the DEEPER is proposed. It uses a combination of bi-directional recurrent neural networks (BRNN) along with Long Short Term Memory (LSTM) as the hidden units to generate a distributed representation for each tuple to capture the similarities between them. Zhang et al. [385] proposed an online Bayesian approach to identify authors with ambiguous names and as a case study, bibliographic data in a temporal stream

format is used and the disambiguation is resolved by partitioning the papers into homogeneous groups.

5.2.3 Graph-based:

As bibliographic data can be viewed as a graph of citations, several approaches have leveraged this property to overcome the problem of author name ambiguation [156, 143, 384, 372]. Hoffart et al. [156] present a method for collective disambiguation of author names, which harnesses the context from a knowledge base and uses a new form of coherence graph. Their method generates a weighted graph of the candidate entities and mentions to compute a dense sub-graph that approximates the best entity-mention mapping. Xianpei et al. [143] aim to improve the traditional entity linking method by proposing a graph-based collective entity linking approach that can model and exploit the global interdependence, i.e., the mutual dependence between the entities. In [384], the problem of author name ambiguity is overcome using relational information considering three graphs: person-person, person-document and document-document. The task becomes then a graph clustering task with the goal that each cluster contains documents authored by a unique real-world author. For each ambiguous name, Xu et al. [372] build a network of papers with multiple relationships. A network-embedding method is proposed to learn paper representations, where the gap between positive and negative edges is optimized. Further, HDBSCAN is used to cluster paper representations into disjoint sets such that each set contains all papers of a unique real-world author.

5.3 Approach:

In this paper, AND is designed using a bibliographic dataset $\mathcal{D} = \{d_i\}_{i=1}^{N}$, consisting of N bibliographic records, where each record d_i refers to a unique publication such that $d_i = \{t_i, s_i, \langle a_{i,u}, \delta_{i,u} \rangle_{u=1}^{\omega_i}\}$. Here, t_i and s_i denote the *title* and *source* of the record, respectively. $a_{i,u}$ and $\delta_{i,u}$ refer to the uth author and its corresponding name, respectively, among ω_i co-authors of d_i. Let $\Delta = \{\delta(m)\}_{m=1}^{M}$ be a set of M unique author names in D shared by a set of L unique authors $\mathcal{A} = \{a(l)\}_{l=1}^{L}$ co-authoring all records in D, where $L >> M$. Note that each author name $\delta(m)$ might refer to one or more authors in $\mathcal{A}$ and each author $a(l)$ might be referred to by one or two author names in Δ. This is because we consider two variates for each author as it might occur differently in different papers. For example, the author "*Rachid Deriche*" is assigned to two elements in Δ, namely "*Rachid Deriche*" and "*R. Deriche*".

Given a reference record $d^* \notin \mathcal{D}$, the goal of our approach is to link each author name $\delta_u^* \in \Delta$ that occurs in d^* to the appropriate author in $\mathcal{A}$ by leveraging t^*, s^* and $\{\delta_u^*\}_{u=1}^{\omega^*}$. Figure 13 illustrates an overview of

our proposed approach. First, the approach computes the correspondence frequency $\delta_u^* \mathbf{R}\mathcal{A}$ that returns the number of authors in $\mathcal{A}$ corresponding to δ_u^*. $\delta_u^* \mathbf{R}\mathcal{A} = 0$ indicates that δ_u^* corresponds to a new author $a(\text{new}) \notin \mathcal{A}$. $\delta_u^* \mathbf{R}\mathcal{A} = 1$ indicates that δ_u^* corresponds to only one author $a(l) \in \mathcal{A}$. In this case, we directly assign δ_u^* to $a(l)$ and no further processing is necessary. Note that in this case, δ_u^* might also refer to a new author $a(\text{new}) \notin \mathcal{A}$ who has the same name as an existing author $a(l) \in \mathcal{A}$. However, our approach does not handle this situation. Please refer to Section 5.4.3 that lists the limitation of the proposed approach.

The goal of this paper is to handle the case of $\delta_u^* \mathbf{R}\mathcal{A} > 1$ which indicates that δ_u^* can refer to more than one author. To this end, the approach extracts the atomic name variate from the author name δ_u^*. For example, for the author name $\delta_u^* =$ "*Lei Wang*", the atomic name variate is $\overline{\delta_u^*} =$ "*L Wang*". Let $\overline{\delta_u^*}$ correspond to $\overline{\delta_\mu}$ which denotes the μth atomic name variate among K possible name variates. Afterwards, the corresponding Neural Network model $\theta_\mu \in \Theta = \{\theta_k\}_{k=1}^{K}$ is picked to distinguish between all authors $\mathcal{A}_\mu = \{a(l_\mu)\}_{l_\mu=1}^{L_\mu}$ who share the same name variate $\overline{\delta_\mu}$.

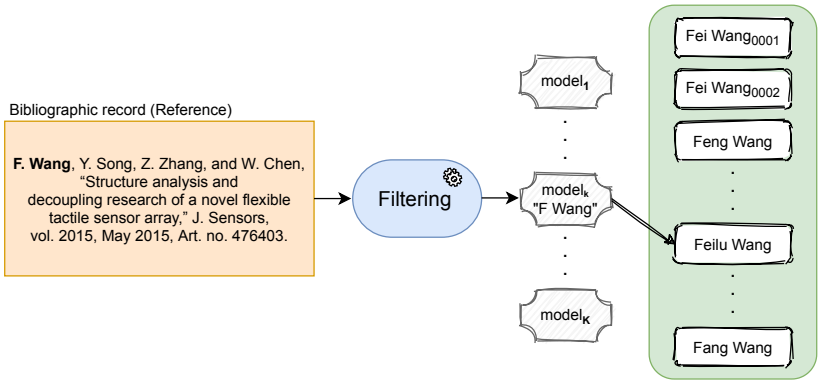


Figure 13: An illustration of the task for linking a name mentioned in the reference string with the corresponding DBLP author entity.

5.3.1 Model Architecture

The Neural Network (NN) model θ_μ takes as input the attributes of d^*, namely the first name of the target author $\delta_u^{*\text{first-name}}$, full names of two co-authors δ_p^* and δ_j^*, title t^* and source s^*. Figure 14 illustrates the architecture of θ_μ, with an output layer of length L_k corresponding to the number of unique authors in $\mathcal{A}_\mu$ who have the same atomic name variate δ_k. As shown in Figure 14, θ_μ takes two inputs $\mathbf{x}_{\mu,\mathbf{1}}$ and $\mathbf{x}_{\mu,\mathbf{2}}$, such that:

$$\mathbf{x}_{\mu,\mathbf{1}} = \text{char2vec}(\delta_u^{*\text{first-name}}) \bigoplus \frac{1}{2}\left(\text{char2vec}(\delta_p^*) + \text{char2vec}(\delta_j^*)\right), \qquad \mathbf{x}_{\mu,\mathbf{2}} = \frac{1}{2}\left(\text{bert}(t^*) + \text{bert}(s^*)\right), \tag{34}$$

where char2vec($\mathbf{w}$) returns a vector representation of length 200 generated using *Char2Vec* [62], which provides a symbolic representation of w. bert($\mathbf{w}$) returns a vector representation of each token in $\mathbf{w}$ w.r.t its context in the sentence. This representation of length 786 is generated using BERT [95]. The goal of separating the two inputs is to overcome the sparseness of content embedding and force the model to emphasise more on target author representation.

All the hidden layers possess a ReLU activation function, whereas the output is a Softmax classifier. Since the model has to classify thousands of classes, each of which is represented with very few samples, 50% of the units in the last hidden layers are dropped out during training to avoid overfitting. Furthermore, the number of publications significantly differs from one author to another. Therefore, each class (i.e. the author) is weighted according to its number of samples (i.e. publications). The model is trained with *adam* optimizer and sparse categorical cross-entropy loss function. Our empirical analysis showed that the best performance was achieved with this architecture and these parameters, which were obtained through grid search.

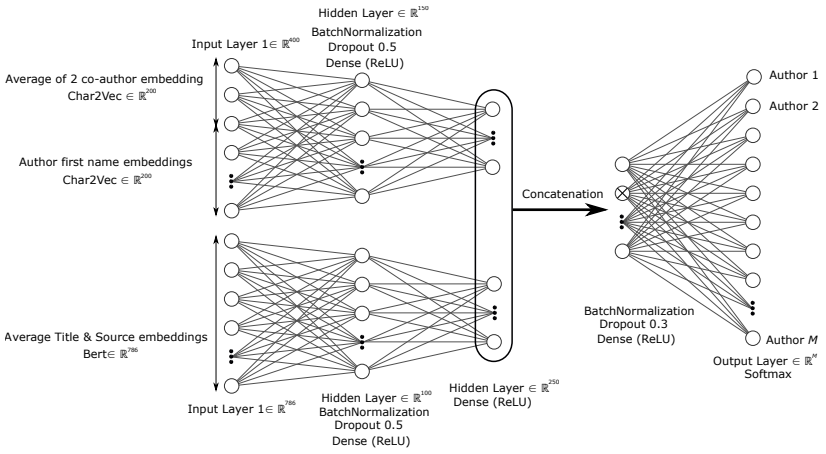


Figure 14: The architecture of our model.

5.3.2 Author name representation

The names of authors do not hold any specific semantic nature as they are simply a specific sequence of characters referring to one or more persons. Therefore, we need a model that can encode words based on the order and distribution of characters such that author names with a similar name

spellings are encoded closely, assuming possible manual editing errors of cited papers.

Chars2vec is a powerful NN-based language model that is preferred when the text consists of abbreviations, typos, etc. It captures the non - vocabulary words and places words with similar spelling closer in the vector space. This model uses a fixed list of characters for word vectorization, where a one-hot encoding represents each character.

5.3.3 Source and Title embedding

The source (e.g. journal names and book titles) of reference can provide a hint about the area of research of the given reference. In addition, the title is a meaningful sentence that embeds the specific topic of the reference. Therefore, we used these two features to capture the research area of the author. Contrary to the author's name, the goal here is to capture the context of the sequences of words forming the title and source. Therefore, we employed the pre-trained BERT model [95] to obtain sentence embeddings of both the title and source.

5.3.4 Model Training

Given the training set $\mathcal{D}_\mu \subset \mathcal{D}$ that corresponds to the subset of bibliographic records authored by authors having the atomic name variate $\overline{\delta_\mu}$, $d_{i_\mu} \in \mathcal{D}_\mu$ generates ω_{i_μ} training samples $\langle \delta_\mu, \delta_{i_\mu,p}, \delta_{i_\mu,j}, t_{i_\mu}, s_{i_\mu} \rangle_{p=1}^{\omega_{i_\mu}}$, where $\delta_{i_\mu,j}$ is a random co-author of d_{i_μ} and might be also the same author name as $\delta_{i_\mu,p}$ and/or δ_μ. Note also that we consider one combination where $\delta_{i_\mu,p} = \delta_\mu$. In order to train the model with the other common name variate where the first name is substituted with its initial, for each sample, we generate another version with name variates $\langle \overline{\delta_\mu}, \overline{\delta_{i_\mu,p}}, \overline{\delta_{i_\mu,j}}, t_{i_\mu}, s_{i_\mu} \rangle$. Consequently, each bibliographic record is fed into the model $2 \times \omega_{i_\mu}$ times.

Since the third co-author $\delta_{i_\mu,p}$ is randomly assigned to the training sample among ω_{i_μ} co-authors d_{i_μ}, we randomly reassign it after Y epochs. In addition to lower training complexity, this has shown in the conducted experiments a slightly better result than training the model at each epoch with samples of all possible co-author pairs p and j.

5.3.5 Model Tuning

For each training epoch, *WhoIs* model fine-tunes the parameters to predict the appropriate target author. The performance of the model is considerably influenced by the number of epochs set to train. Specifically, a low epoch count may lead to underfitting. Whereas, a high epoch count may lead to over-fitting. To avoid this, we enabled early stopping, which allows the model to specify an arbitrarily large number of epochs.

Keras supports early stopping of the training via a callback called *EarlyStopping*. This callback is configured with the help of the *monitor* argument which allows setting the validation loss. With this setup, the model receives a trigger to halt the training when it observes no more improvement in the validation loss.

Often, the very first indication of no more improvement in the validation loss would not be the right epoch to stop training; because the model may start improving again after passing through a few more epochs. We overcome this by adding a delay to the trigger in terms of consecutive epochs count on which, we can wait to observe no more improvement. A delay is added by setting the *patience* argument to an appropriate value. *patience* in *WhoIs* is set to 50, so that the model only halts when the validation loss stops getting better for the past 50 consecutive epochs.

5.3.6 Model checkpoint

Although *WhoIs* stops the training process when it achieves a minimum validation loss, the model obtained at the end of the training may not give the best accuracy on validation data. To account for this, Keras provides an additional callback called *ModelCheckpoint*. This callback is configured with the help of another *monitor* argument. We have set the *monitor* to monitor the validation accuracy. With this setup, the model updates the weights only when it observes better validation accuracy compared to earlier epochs. Eventually, we end up persisting in the best state of the model with respect to the best validation accuracy.

5.3.7 Prediction:

Given the new bibliographic record $d^* = \{t^*, s^*, \langle \delta_u^* \rangle_{u=1}^{\omega^*}\}$, the goal is to disambiguate the author name δ_{target}^* which is shared by more than one author ($\delta_{\text{target}}^* \mathbf{R}\mathcal{A} > 1$). To this end, Y samples $S_{y=1}^{Y}$ are generated for all possible pairs of co-author names p and j: $\langle \delta_{\text{target}}^*, \delta_p^*, \delta_j^*, t^*, s^* \rangle_{p=1,j=1}^{\omega^*,\omega^*}$, where $Y = \text{C}(\omega^* + 1, 2)$, i.e. the combination of $\omega^* + 1$ authors taken 2 at a time, and δ_u^* can be a full or abbreviated author name. All the Y samples are fed to the corresponding model θ_μ, where the target author a_{target} of the target name δ_{target}^* is predicted as follows:

$$a_{target} = \underset{1 \cdots L_\mu}{argmax} \left(\theta_\mu(S_1) \oplus \theta_\mu(S_2) \oplus \cdots \oplus \theta_\mu(S_Y) \right), \tag{35}$$

where $\theta_\mu(S_y)$ returns a probability vector of length L_μ with each element l_μ denotes the probability of the author name δ_{target}^* to be the author a_{l_μ}.

5.4 Experiments

This section presents the experimental results of the proposed approach to the DBLP dataset.

5.4.1 Dataset

The following datasets are widely used to evaluate author name disambiguation approaches but the results on these datasets cannot reflect the results on real scenario streaming data.

- **ORCID** [25]**:** it is the largest accurate dataset as the publication is assigned to the author only after authorship claim or another rigorous authorship confirmation. However, this accuracy comes at the cost of the number of assignments. Our investigation shows that most of the registered authors are not assigned to any publication and an important number of authors are not even registered. This is because most of the authors are not keen to claim their publications due to several reasons.
- **KDD Cup 2013** [26]**:** it is a large dataset that consists of 2.5M papers authored by 250K authors. All author metadata are available including affiliation.
- **Manually labelled (e.g. PENN** [27]**, QIAN** [28]**, AMINER** [29]**, KISTI** [30]**):** These datasets are supposed to be very accurate since they are manually labelled. However, this process is expensive and time-consuming and, therefore, it can cover only a small portion of authors who share the same names.

In this work, we collected our dataset from the DBLP bibliographic repository[31]. The DBLP version of July 2020 contains 5.4 million bibliographic records such as conference papers, articles, thesis, etc., from various fields of research. As stated by the maintainers of DBLP [32], the accuracy of the data is not guaranteed. However, a lot of effort is put into manually disambiguating homonym cases when reported by other users. Consequently, we are aware of possible homonym cases that are not resolved yet. From the repository, we collected only records of publications published in journals and

[25] https://figshare.com/articles/ORCID_Public_Data_File_2017/5479792

[26] https://www.kaggle.com/c/kdd-cup-2013-author-paper-identification-challenge

[27] http://clgiles.ist.psu.edu/data/nameset_author-disamb.tar.zip

[28] https://github.com/yaya213/DBLP-Name-Disambiguation-Dataset

[29] http://arnetminer.org/lab-datasets/disambiguation/rich-author-disambiguation-data.zip

[30] http://www.lbd.dcc.ufmg.br/lbd/collections/disambiguation/DBLP.tar.gz/at_download/file

[31] https://dblp.uni-trier.de/xml/ (July 2020)

[32] https://dblp.org/faq/How+accurate+is+the+data+in+dblp.html

proceedings. Each record in this collection represents metadata information of a publication with one or more authors, title, journal, year of publication and a few other attributes. The availability of these attributes differs from one reference to another. Also, the authors in DBLP who share the same name have a suffix number to differentiate them. For instance, the authors with the same name 'Bing Li' are given suffixes such as 'Bing Li 0001', and 'Bing Li 0002'. The statistical details of the used DBLP collection are shown in Table 22.

Table 22: Statistical details of the used DBLP collection.

# of records	5258623
# of unique authors	2665634
# of unique author names	2613577
# of unique atomic name variates	1555517

Figure 15 indicates that the majority of target authors in the sub-collections (each sub-collection includes all records of authors with the same name) have distinct full names. However, a considerable number of them share full names, leading to a significant challenge, particularly when multiple authors (e.g. over 80 in 4 out of 5 sub-collections) share the same full name but have an unequal number of publications. In such cases, it becomes more challenging to differentiate these authors from the dominant author with the same name.

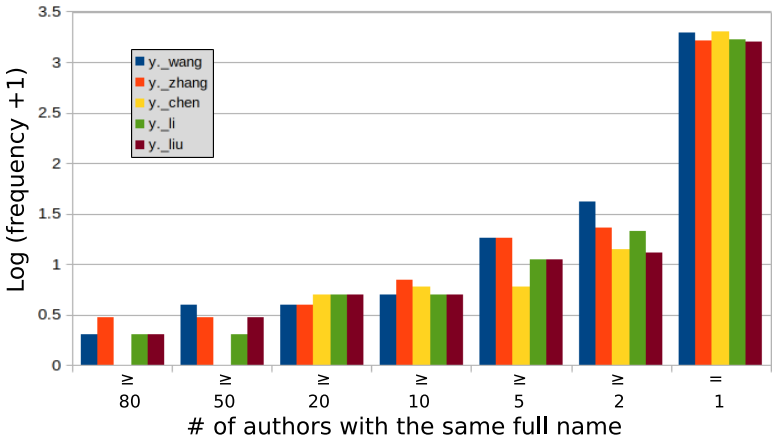


Figure 15: The *log* frequency of authors sharing the same full name for the top five sub-collections.

Figure 16 illustrates the log frequency of bibliographic records with the same full name in the top five sub-collections used in this paper. As illustrated, in all sub-collections, the target authors of around half of the records authored a few records (less than 5) and have unique names. Although it is simple to distinguish these authors when their full names occur, it is ex-

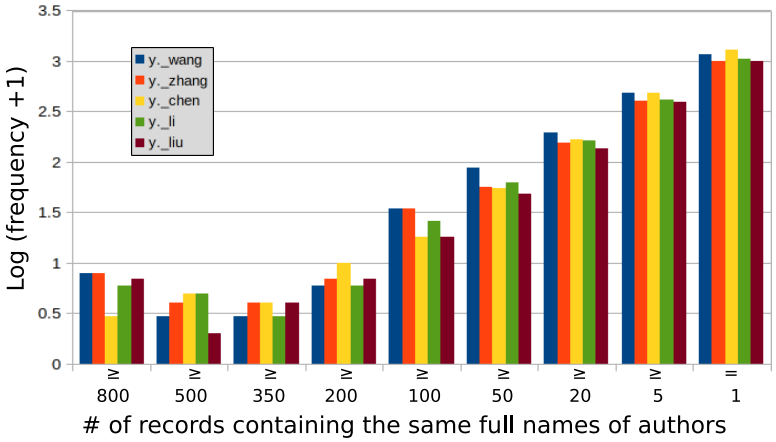


Figure 16: The *log* frequency of records with the same full name of the target author for the top five sub-collections.

tremely challenging to recognize them among more than 2000 authors sharing the same atomic name variate due to the unbalance of records with the other authors.

Figure 17 shows the frequency of authors sharing the same names and the same atomic name variates. As can be seen, the problem is more critical when the authors are cited with their atomic name variate as there are five atomic name variates shared by around $11.5k$ authors. This makes the problem of disambiguation critical because not only targets authors who might share the same atomic name variate but also their co-authors. For instance, we observed publications authored by the pair of co-authors having the atomic name variates: *Y. Wang* and *Y. Zhang*. However, they refer to different *Y. Wang* and *Y. Zhang* pairs of real-world authors.

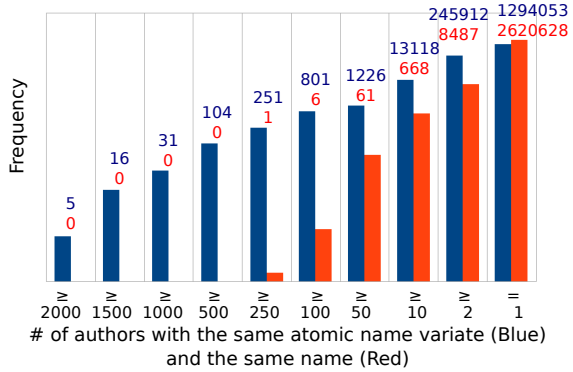


Figure 17: Frequency of authors sharing the same atomic name variate (Blue) / the same full name (Red).

Since our approach gathers authors with the same name variates, 261464 models are required to disambiguate all author names in our collection. Therefore, we present in this paper the experimental results on 5 models

corresponding to the highest number of authors sharing the same name variates. Table 23 presents statistical details of the five sub-collections which demonstrates the challenges inherent in author name disambiguation in real-world scenarios. # **R2A** for instance, in some publications, two co-authors have the same exact names. This makes the disambiguation more difficult as these authors share not only their names but also co-authors and papers.

Table 23: Statistical details of the top 5 sub-collections of authors sharing the same atomic name variates, where # **ANV** is the corresponding atomic name variate, # **UTA** is the number of unique target authors, # **RCD** is the number of bibliographic records, # **UCA** is the number of unique co-author full names, # **UAN** is the number of unique target author full names, # **R2A** is the number of records with two co-authors of the same record having the same names or the same atomic name variates and # **R3A** is the number of records with three co-authors of the same record having the same names or the same atomic name variates. For # **R2A** and # **R3A**, it is not necessary that the authors have the same name / atomic name variate as the target author but most probably.

	'Y Wang'	'Y Zhang'	'Y Chen'	'Y Li'	'Y Liu'
# **UTA**	2601	2285	2260	2166	2142
# **RCD**	37409	33639	26155	29154	27691
# **UCA**	43199	39389	33461	35765	33754
# **UAN**	2005	1667	2034	1734	1606
# **R2A**	582	598	316	372	338
# **R3A**	13	12	4	4	3

To ensure a credible evaluation and result reproducibility in real scenarios, we split the records in each sub-collection into a training set ($\sim$ 70%), validation set ($\sim$ 15%) and testing set ($\sim$ 15%) in terms of records/target author. Specifically, for each target author, we randomly split the corresponding records. If the target author did not author enough publications for the split, we prioritize the training set, then validation and finally the test set. Consequently, the number of samples is not necessarily split according to $70 : 15 : 15$ as the number of co-authors differs among publications. Moreover, it is highly likely that the records of a unique target author are completely different among the three sets. Consequently, it is difficult for the model to recognize the appropriate author only from his/her co-authors and research area. However, we believe that this is more realistic and a perfect simulation of the real scenario.

To account for possible name variates, each input sample of full names is duplicated, where the duplicate down sample full names of all co-authors to atomic name variates. Note that this is applied to training, validation and test sets. The goal is to let the model capture all name variates for each author and his/her co-authors. In none of the sets, the variates are mixed in a single sample as we assume that this case is very less likely to occur

in the real world. The experiments were conducted on a machine with the following specifications:

- Processor: AMD Ryzen Threadripper 1950X 16-Core
- RAM: 12 GB
- Graphics card: NVIDIA Titan V GV100

The algorithm was implemented in Python 3.7 using the TensorFlow library.

5.4.2 Results

The existing AND approaches use different datasets to design and evaluate their models. This lead to different assumptions and challenge disparity. Unfortunately, the codes to reproduce the results of these approaches are not available or easily accessed [168]. Therefore, it is not possible to fairly compare *WhoIs* against baseline approaches. For future work, our code and the used datasets are publicly available [33].

Table 24 presents the result of *WhoIs* on the sub-collections presented in Table 23. The label *All* in the table denotes that all samples were predicted twice, one with full names of the target author and its co-authors and another time with only their atomic name variates, whereas the label *ANV* denotes that only samples with atomic names are predicted. The obtained results show that an important number of publications are not properly assigned to their appropriate authors. This is due to the properties of the sub-collections which were discussed above and statistically presented in Table 23. For example, 1) two authors with the same common name authoring a single publication. 2) more than one author with the same common atomic name variate authoring a single publication, 3) number of authors with the same full name, 4) the uncertainty of the accuracy of the dataset, etc.

Although the comparison is difficult and cannot be completely fair, we compare *WhoIs* to other state-of-the-art approaches, whose results are reported in [384]. These results are obtained on a collection from CiteSeerX [34] that contains records of authors with the name / atomic name variate '*Y Chen*'. This collection consists of 848 complete documents authored by 71 distinct authors. We picked this name for comparison because of two reasons; 1) the number of authors sharing this name is among the top five as shown in Table 23 and 2) All methods cited in [384] could not achieve a good result. We applied *WhoIs* on this collection by randomly splitting the records into 70% for training, 15% for validation and 15% for testing. The results are shown in Table 25. Note that in our collection, we consider way more records and distinct authors (see Table 23) and we use only reference attributes (i.e. co-authors, title and source).

[33] `https://doi.org/10.5281/zenodo.7744775`
[34] `http://clgiles.ist.psu.edu/data/`

Table 24: Detailed results of *WhoIs* on the sub-collections corresponding to the top five of authors sharing the same atomic name variates in the DBLP repository. The results are presented in terms of Micro average precision (**MiAP**), Macro average precision (**MaAP**), Micro average recall (**MiAR**), Macro average recall (**MaAR**), Micro average F1-score (**MiAF1**) and Macro average F1-score (**MaAF1**). **ANV** denotes that only atomic name variates were used for all target authors and all their co-authors.

	'Y Wang'	'Y Zhang'	'Y Chen'	'Y Li'	'Y Liu'
MaAP(ANV)	0.226	0.212	0.255	0.193	0.218
MaAP(All)	0.387	0.351	0.404	0.342	0.347
MaAR(ANV)	0.299	0.276	0.301	0.229	0.267
MaAR(All)	0.433	0.383	0.409	0.339	0.361
MaAF1(ANV)	0.239	0.220	0.258	0.195	0.223
MaAF1(All)	0.385	0.342	0.383	0.321	0.332
MiAF1(ANV)	0.274	0.278	0.366	0.260	0.322
MiAF1(All)	0.501	0.482	0.561	0.492	0.504

As the results presented in Table 25 show, *WhoIs* outperforms other methods in resolving the disambiguation of the author name '*Y Chen*' on the CiteSeerX dataset, which is a relatively small dataset and does not really reflect the performance of all presented approaches in real scenarios. The disparity between the results shown in Table 24 and Table 25 demonstrates that the existing benchmark datasets are manually prepared for the sake of accuracy. However, this leads to covering a very small portion of records whose authors share similar names. This disparity confirms that author name disambiguation is still an open problem in digital libraries and far from being solved.

The obtained results of *WhoIs* illustrate the importance of relying on the research area of target authors and their co-authors to disambiguate their names. However, they trigger the need to encourage all authors to use different author identifiers such as ORCID [26] in their publications as the automatic approaches are not able to provide a perfect result mainly due to the complexity of the problem.

5.4.3 Limitations and obstacles of *WhoIs*:

WhoIs demonstrated a satisfactory result and outperformed state-of-the-art approaches on a challenging dataset. However, the approach faces several obstacles that will be addressed in our future works. In the following, we list the limitations of the proposed approach:

- New authors cannot be properly handled by our approach, where a confidence threshold is set to decide whether the input corresponds to a new

Table 25: Comparison between *WhoIs* and other baseline methods on CiteSeerX dataset in terms of Macro F1 score as reported in [384]. **ANV** denotes that only atomic name variates were used for all target authors and all their co-authors.

	Macro ALL/ANV	Micro ALL/ANV
WhoIs	**0.713** / **0.702**	0.873 / 0.861
NDAG [384]	0.367	N/A
GF [198]	0.439	N/A
DeepWalk [280]	0.118	N/A
LINE [336]	0.193	N/A
Node2Vec [132]	0.058	N/A
PTE [335]	0.199	N/A
GL4 [151]	0.385	N/A
Rand [384]	0.069	N/A
AuthorList [384]	0.325	N/A
AuthorList-NNMF [384]	0.355	N/A

author or an existing one. To our knowledge, none of the existing supervised approaches is capable to handle this situation.

- Commonly, authors found new collaborations which lead to new co-authorship. Our approach cannot benefit from the occurrence of new co-combinations of co-authors as they were never seen during training.
  Planned solution: We will train an independent model to embed the author's discipline using his/her known publications. With this, we assume that authors working in the same area of research will be put close to each other even if they did not publish a paper together, the model would be able to capture the potential co-authorship between a pair of authors in terms of their area of research.

- Authors continuously extend their research expertise by co-authoring new publications in relatively different disciplines. This means that the titles and journals are not discriminative anymore. Consequently, it is hard for our approach to disambiguate authors holding common names.
  Planned solution: we plan to determine the author's areas of research by mining domain-specific keywords from the entire paper instead of its title assuming that the author uses similar keywords/writing styles even in different research areas with gradual changes which can be captured by the model.

- There are a lot of models that have to be trained to disambiguate all authors in the DBLP repository.

- Commonly, the number of samples is very small compared to the number

of classes (i.e. authors sharing the same atomic name variate) which leads to overfitting the model.
Planned solution: we plan to follow a reverse strategy of disambiguation. Instead of employing the co-authors of the target author, we will employ their co-authors aiming to find the target author among them. We aim also to learn co-author representation by employing their co-authors to help resolve the disambiguation of the target author's name.

- As mentioned earlier and stated by the maintainers of the platform [35], the accuracy of the DBLP repository is not guaranteed.

5.5 Conclusion

We presented in this paper a comprehensive overview of the problem of AND. To overcome this problem, we proposed a novel framework that consists of a lot of supervised models. Each of these models is dedicated to distinguishing among authors who share the same atomic name variate (i.e. first name initial and last name) by leveraging the co-authors and the titles and sources of their known publications. The experiments on challenging and real-scenario datasets have shown promising and satisfactory results on AND. We also demonstrated the limitations and challenges that are inherent in this process.

To overcome some of these limitations and challenges, we plan for future work to exploit citation graphs so that author names can be linked to real-world entities by employing the co-authors of their co-authors. We assume that using this reverse process, the identity of the target author can be found among the co-authors of his/her co-authors. We plan also to learn the research area of co-authors in order to overcome the issue of new co-authorships.

[35] https://dblp.org/faq/How+accurate+is+the+data+in+dblp.html

6 Paper 7: ICD Coding with Knowledge Graph

Knowledge Guided Multi-filter Residual Convolutional Neural Network for ICD Coding from Clinical Text

Zeyd Boukhers(✉), *Prantik Goswami, Jan Jürjens*

(DOI: 10.1109/10.13026/ C2XW26)

Abstract One challenge often encountered when using Deep Neural Network models for automatic ICD coding is their potential inability to effectively handle unseen clinical texts, especially when these models are only trained on a limited number of examples. This is because these models rely solely on the patterns and relationships present in the training data, and may not be able to effectively incorporate additional knowledge about the relationships between medical entities. To address this issue, we introduce *KG-MultiResCNN - **K**nowledge **G**uided **Multi**-filter **Res**idual **C**onvolutional **N**eural **N**etwork* model, which combines training examples with external knowledge from the Wikidata Knowledge Graph (KG) in order to better capture the relationships between medical entities. The KG is a structured database that contains a wealth of information about various entities, including medical concepts and their relationships with one another. By incorporating this external knowledge into our model, we are able to improve its ability to predict ICD codes for new clinical texts. In our experiments with the MIMIC-III dataset, we found that the KG-MultiResCNN model significantly outperformed the baseline approaches. This demonstrates the effectiveness of using external knowledge, in addition to training examples, to improve the performance of deep learning models for automatic ICD coding.

Keywords: *ICD Coding, Automatic Diagnosis, CNN*

6.1 Introduction

In the past decade, Deep Learning (DL) and Natural Language Processing (NLP) techniques have been widely used in healthcare research [366, 258, 255, 256, 117, 229, 223] due to a large amount of health data available. One significant application of these techniques is in medical diagnostic decision-making [207, 332], as deep learning approaches applied to medical images have already achieved accuracy on par with human professionals. DL techniques applied to textual data, such as Electronic Health Records (EHR), are also gaining attention, particularly for the automatic detection and assignment of International Classification of Diseases (ICD) codes. The ICD is a globally recognized list of codes developed and maintained by the World Health Organization (WHO) to represent diagnoses and medical procedures

with universal codes for healthcare systems such as hospitals and health insurance companies. It is commonly used by healthcare providers for a variety of purposes, including improving the usability and maintainability of records, facilitating reimbursement, and enabling the storage and retrieval of diagnostic and procedural information whenever needed [45, 251]. As part of hospital services, clinical EHRs are often linked to the corresponding ICD codes for each patient's hospital admission, allowing for better organization and management of patient data.

The use of automatic ICD coding from textual clinical notes has been a topic of research for over two decades [206, 90]. Early methods often relied on manually-created features [308], but as technology and data processing power have improved, a range of approaches have been developed. Perotte et al. [279] used a Support Vector Machine (SVM) to classify "flat" and "hierarchical" ICD codes, while Koopman et al. [193] also used an SVM to classify hierarchical ICD codes related to cancer from textual death certificates. Shi et al. [315] used a character-level Long Short-Term Memory (LSTM) model to identify similarities between discharge summary notes and ICD code descriptions. Prakash et al. [284] developed a neural memory network model called "C-MemNNs" that learned representations from textual data and predicted top-50 and top-100 codes and also used Wikipedia as an external knowledge to improve model performance. Vani et al. [349] created a Grounded Recurrent Neural Network (GRU) that utilized label-specific dimensions for hidden units to predict specific diseases. Baumel et al. [32] used a Hierarchical Attention-Bidirectional Gated Recurrent Unit (HA-GRU) to assign multiple ICD codes to patients' discharge summary notes. Wang et al. [354] proposed a mixed embedding model that calculated the cosine similarity between word embedding vectors and label vectors in the same embedding space to predict the labels.

Li and Yu [212] recently proposed the Multi-Filter Residual Convolutional Neural Network (MultiResCNN) as a state-of-the-art model for predicting multiple possible ICD codes from the content of the discharge summaries. The model uses multiple filter CNN networks followed by residual networks and was evaluated on the MIMIC-III discharge summary notes dataset, where it achieved satisfactory results. However, like many other existing approaches, the model still struggles to effectively capture the correlation between diseases (represented by ICD codes) and the physiological and symptom attributes mentioned in clinical text. This is a significant challenge because most current methods rely only on training examples (i.e. clinical cases documented in clinical texts) to learn this correlation. However, the high dimensionality and sparsity of the feature/class space make it difficult to find a sufficient number of training examples, in reality, to accurately model this relationship. The dimensionality refers to the number of possible diseases and physiological, symptom, and lab-test attributes, while sparsity refers to the rarity of certain attributes in clinical cases. As a result,

there is a need for more effective methods that can better handle the high dimensionality and sparsity of this task, and further improve the accuracy of automatic ICD coding from free-text clinical notes.

The goal of this research is to improve the state-of-the-art method for automatic ICD coding from clinical texts, which currently struggles to effectively capture the relationship between diseases and physiological and symptom attributes mentioned in the text. To address this issue, the proposed approach simulates the way physicians interpret clinical texts into diagnoses, using their medical knowledge to understand the clinical situation and the relationships between different diseases, symptoms, and treatments. Consequently, this approach aims to improve the performance of automatic ICD coding by incorporating external medical knowledge in the form of a knowledge graph. To this end, this work enhances the state-of-the-art method proposed by Li and Yu [212] by guiding the model with external medical knowledge. To incorporate this structured medical knowledge into the model, we introduce *KG-MultiResCNN - **K**nowledge **G**uided **Multi**-filter **Res**idual **C**onvolutional **N**eural **N**etwork* model that is guided by an additional embedding vector. This vector is a knowledge graph embedding of medical entities automatically extracted from the clinical text and is concatenated with the word embedding vector. The model is then trained using both the original text word embeddings and the knowledge graph embeddings. We also compute the Term Frequency-Inverse Document Frequency (TF-IDF) value for each word in the clinical text as a weighting factor for the medical entities and use two residual (ResNet) blocks to extract better feature representation due to the large size of the embedding vectors. The assumption is that medical entities that are not synonyms and have similar relationships should have similar embeddings. Overall, this work aims to tackle a single but important research question: "*Does the inclusion of knowledge graph support the process of automatic ICD coding?*". The main contributions of this work are as follows:

- Improved the MultiResCNN [212] model by introducing an additional embedding layer based on a knowledge graph of significant medical entities extracted from the text;
- Used knowledge graph embedding for automatic ICD coding for the first time, to our knowledge;
- Weighted the importance of each word in the text using the Term Frequency-Inverse Document Frequency (TF-IDF) score as a weighting factor;
- Employed two residual (ResNet) blocks to improve feature representation and handle the large size of the embedding vectors;
- Made all implementations publicly available for further research.

The remainder of this paper is organized as follows: In Section 6.2, we review previous research on the topic and discuss the relevant approaches and their strengths and limitations. Section 6.3 describes our proposed method in detail with all technical details. Section 6.4 presents the results of the experimental evaluation of our method, including statistical analyses and comparisons with other approaches. Finally, in Section 6.5, we summarize the main findings of our research and discuss the implications of the results for future work. We also include recommendations for practical applications and directions for future research.

6.2 Related Work

Assigning an ICD code to a free-text EHR document is a challenging and arduous process. It demands expertise in the healthcare field and can be both financially and error-prone. This has led to prolonged research on developing automatic methods to extract ICD codes from clinical notes for over two decades [206, 90]. In this section, we thoroughly review the most critical ICD coding techniques, grouping them into three distinct categories for enhanced comprehension and organization.

6.2.1 Classical Machine Learning

Early efforts to assign ICD codes to inpatient episodes have largely relied on manually crafted features [308] and traditional machine learning models. Perotte et al. [279] used a support vector machine (SVM) to classify flat and hierarchical ICD codes, while Koopman et al. [193] employed a similar SVM approach to classifying hierarchical ICD codes related to cancer from free-text death certificates. Ferrao et al. [112] proposed an adaptive data processing method that utilizes structured electronic health record data and is trained by SVM classifiers to predict codes, resulting in F1-measure values around 52%. Zhou et al. [396] proposed a regular expression-based approach to establish a correspondence between unique ICD codes and diagnosis descriptions in both outpatient and inpatient settings. Diao et al. [97] evaluated the performance of two feature engineering methods for processing discharge diagnosis and procedure texts, using the gradient boosting algorithm on a dataset of 71,709 admissions at Fuwai Hospital and 168 primary diagnoses with ICD-10 codes.

6.2.2 Neural Network-based approaches

Over the past decade, the majority of proposed ICD coding solutions have been based on Neural Networks, such as in [315, 349, 212], due to their impressive performance across a variety of tasks. Shi and colleagues ([315]) utilized character-level LSTM to identify similarities between discharge summary notes and ICD code descriptions. Vani et al. [349] developed a Grounded

Recurrent Neural Network (GRU) that incorporates label-specific dimensions for hidden units to predict specific diseases. Baumel et al. [32] employed a Hierarchical Attention-bidirectional Gated Recurrent Unit (HA-GRU) to assign multiple ICD codes to patients' discharge summary notes. Wang et al. [354] proposed a mixed embedding model, assuming that projecting word and label vectors in the same embedding vector space would lead to better results. Their model calculates the cosine similarity between word embedding vectors and label vectors to predict the labels. Xu et al. [373] proposed an ensemble-based approach that combines the outputs of three neural network models, each handling different types of data (unstructured, semi-structured, and tabular). The models utilize CNNs, LSTMs, and decision trees for data processing and classification. The approach was evaluated using MIMIC-III data and demonstrated improved performance by using multiple modalities of data. Meanwhile, Mullenbach et al [247] proposed the CNN model **CAML**, which utilizes label attention to enhance ICD coding task performance. The model uses pre-trained word vectors and was tested on MIMIC-III and MIMIC-II discharge summary notes, outperforming previous methods.

As the most recent state-of-the-art model, Li and Yu [212] proposed Multi-Filter Residual Convolutional Neural Network (MultiResCNN) which utilizes a one-hot encoded label vector to predict multiple ICD codes related to the discharge summary text. Their approach uses a multiple-filter CNN network, with a residual network [148] following each filter, and employs a label attention mechanism for better prediction accuracy. They evaluated their model on the MIMIC-III discharge summary notes dataset and showed improved performance with both MIMIC-Full codes and MIMIC-50 codes.

The limitation of these approaches is that they rely solely on the examples present in the training set, which can only represent a small subset of the vast and complex space of diseases, symptoms, and epidemiological factors. This can lead to the models being limited in their ability to generalize to new and unseen data. To overcome this limitation, it is crucial to incorporate external knowledge sources that can augment the training data and provide additional information to improve the performance of the models.

6.2.3 Knowledge-enhanced approaches

Many studies have investigated the effect of external information sources on medical text understanding [27, 66, 284]. While Kumar Chanda et al. [66] proposed a method for learning medical term embeddings from limited notes by using medical term definitions as external knowledge, Bai and Vucetic [28] built upon the CAML model by incorporating a Knowledge Source Integration (KSI) framework to improve performance. KSI uses superficial knowledge from Wikipedia to add extra weight to the input text for ICD code prediction, specifically focusing on rare diseases. The model was evaluated

on the MIMIC-III dataset and showed improved performance in predicting rare diseases. These studies demonstrated the need for external knowledge, but the unstructured knowledge used can be difficult for the machine to process. As an alternative, it may be beneficial to incorporate structured knowledge sources in the form of knowledge graphs.

Choi et al. [77] introduced **GRAM**, which combines information from medical ontologies with deep learning models via attention mechanism. Ancestors of less frequent medical concepts are adaptively combined by frequency and attention, and the attention mechanism is trained end-to-end. This means that if enough training data are available, **GRAM** achieves comparable results without incorporating the medical ontology. In contrast, **KAME** [230] exploits a medical ontology (i.e. ICD 9) to learn representations of medical codes and their ancestors in the whole prediction process. Bao et al. [31] used ICD descriptions as external knowledge sources to improve medical code prediction in their hybrid capsule network model with a bi-directional LSTM and label embedding framework. Similarly, Du et al. [100] used GCN to obtain diagnosis codes' semantic representations and construct a co-occurrence graph from EHR data, improving token extraction with an attention mechanism to model the interaction between diagnosis codes' ontology representations and clinical notes. Peng et al. [277] proposed MIPO, a healthcare representation learning model that uses medical knowledge and patient journey to predict future diagnoses. MIPO consists of a task-specific representation learning module and a graph-embedding module, and it jointly learns task-specific and ontology-based objectives.

The works mentioned above utilize structured knowledge in the entire prediction process, however, the medical ontologies and ICD descriptions used primarily reveal connections among diseases and not all medical entities mentioned in medical texts, such as symptoms and epidemiological factors. This can hinder the machine's ability to effectively utilize all available medical information and evidence-based knowledge during the prediction process. To address this limitation, a more comprehensive knowledge graph should be properly integrated, which can enable the machine to incorporate a broader range of information and improve the accuracy of predictions.

6.3 KG-MultiResCNN

This paper presents a novel model called *KG-MultiResCNN - **K**nowledge **G**uided **Multi**-filter **Res**idual **C**onvolutional **N**eural **N**etwork*, based on the state-of-the-art approach proposed by Li and Yu [212]. The main contribution of this work is to predict disease ICD codes from unstructured clinical text by leveraging a knowledge graph. The model first extracts tokens from the clinical text and represents them numerically, weighting them according to their importance. Subsequently, it identifies medical entities and represents the relationships between them numerically using knowledge graph

embedding. These representations are concatenated and passed through a **Multi**-filter **Res**idual **C**onvolutional **N**eural **N**etwork to predict the ICD code. We employed CNNs due to their effectiveness in processing sequential unstructured data such as free text. Due to the complexity of the task, a deep CNN is needed. Therefore, residual blocks have been considered to address the vanishing gradient problem. In the following, we discuss each of the elements of *KG-MultiResCNN*:

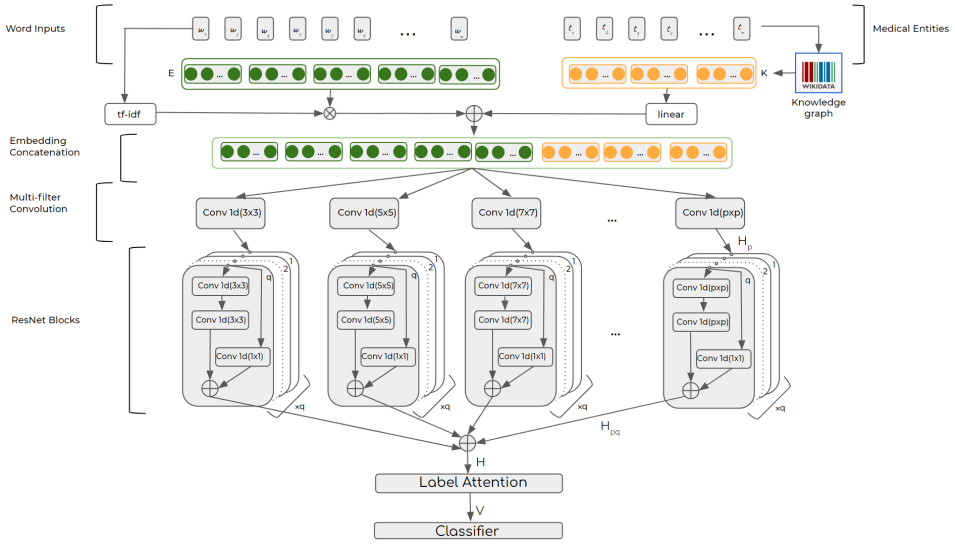


Figure 18: An overview of "*Kg-MultiResCNN*" architecture following the work of Li and Yu[212].

Word Embedding Input

The first part of the input layer is an embedding matrix (E) obtained from the sequence of the words of the text document. The word sequence is denoted as $\mathbf{w}$, which is defined as $\mathbf{w} = (w_1, w_2, ..., w_n)$, where n is the total number of words present in the text. For each word, the embedding vector is obtained using the pretrained word2vec model [237]. Furthermore, each word embedding is weighted using a TF-IDF[36] score. TF-IDF measures the relevance of words such that those frequent in the document but rare in the collection are considered most relevant. Specifically, the embedding vector can be formulated as $\mathbf{e} = g \times \mathbf{u}$ where $\mathbf{u}$ is the word embedding and $g > 0$ is the TF-IDF score of that word. Consequently, the the word embedding input part becomes $E = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\}$ where $e_i \in \mathbb{R}^{d^{(w)}}$. $d^{(w)}$ is the dimension of the word embedding vector.

[36] term frequency-inverse document frequency

Input KG-Embedding Input

The second part of the input layer is the knowledge graph embedding matrix (K), which encodes the relationships between the medical entities present in the clinical text with all related entities regardless of whether they are present in the clinical text or not. To this end, we extract from $\mathbf{w}$ the most significant medical entities using a domain-specific Named Entity Recognition model [37]. This results in the sequence $\mathbf{t}$ denoted as $\mathbf{t} = \{t_1, t_2, ..., t_m\}$, where m is the number of medically significant entities extracted by the entity extraction model. Using each entity j_j, a Knowledge Graph is queried to obtain the knowledge graph embedding k_j. Hence the knowledge graph embedding matrix becomes, $K = \{k_1, k_2, ..., k_m\} \in \mathbb{R}^{m \times d^{(k)}}$, where $d^{(k)}$ denotes the dimension of the knowledge embedding. In this paper, we employed PyTorch BigGraph (PGB)[208] which is an embedding system provided by Meta Research[38] community. PGB learns the node and edges representations of massive knowledge graphs and embeds the nodes and relations in the graph. Its strength lies in the fact that it is trained on the large Wikidata[39] knowledge graph with 78 million entities and 4,131 relations and provides embedding of 200 dimensions. It is highly likely that the medical entities extracted from the clinical text exist in Wikidata and are connected to other medical entities with several relationship types. The word embedding matrix and the KG embedding matrix jointly serve as the input layer (i.e. clinical text representation) to the model.

Multi-Filter Convolution Layer

To map the clinical text representation to the ICD codes, we followed the work of Li and Yu [212] by building a multi-filter 1-dimensional Convolutional Neural Network architecture. The strategy is to pass the varied length of texts through a parallel set of CNN networks. However, the kernel size is of different lengths for each CNN filter. Given p filters, the corresponding kernel size would be k_p and the convolution filter would be $W_p \in \mathbb{R}^{k_p \times d^{(e)} \times d^{(c)}}$ where $d^{(e)}$ is the input dimension and $d^{(c)}$ is the output dimension. In general, the filter/convolution operation on a vector reduces the size of the output vector. However, in this approach, we aim to keep the size of the output vector the same as the input. To this end, the number of parameters is calculated as follows:

$$L_{out} = \left[\frac{L_{in} + 2 \times padding - dilation \times (kernel\_size - 1) - 1}{stride} + 1 \right]$$

[37] `https://huggingface.co/samrawal/bert-base-uncased_clinical-ner`
[38] https://github.com/facebookresearch
[39] https://www.wikidata.org

By setting the stride = 1, dilation = 1, kernel_size = k, and padding = $floor(\frac{k}{2})$, we can achieve our goal of same output size. With all these adjustments, the 1-Dimensional convolution operation can be formalized as :

$$\mathsf{C}_{p,j}(E) = W_p^T \otimes E^{j:j+k_p-1}$$

$$H_p = \sum_{j=1}^{n} tanh(\mathsf{C}_{p,j}(E))$$

Here, $\otimes$ represents a convolution operation and $\mathsf{C}_{p,j}$ indicates the output of p^{th} convolution where the input matrix position starts from jth row and ends at the row $j + k_p - 1$. H_n indicates the final layer output after the convolution output is passed though $tanh$ activation for total n sequence of input and then concatenated (indicated by $\sum$) together.

Residual Convolution Layer

The output of each convolutional filter again goes through a series of convolution filters called a residual block. Each of these blocks consists of 3 convolution layers. A typical 1-D convolution architecture is shown in Figure 19, where the convolution filter W_p slides through the embedding matrix E with a stride of 1. Formally, if we consider p multi-filter convolution layers then each of these convolution filters has a series of q residual blocks on top. Each of the residual blocks have 3 convolution filters, namely $r_{pq_1}, r_{pq_2}, r_{pq_3}$ and their corresponding filter weights are $W_{pq_1}, W_{pq_2}, W_{pq_3}$, where r_{pq} is the q^{th} residual block on top of p^{th} multi-filter convolution layer. The output of each convolution filter inside a residual block can be formulated as

$$\mathsf{C}_{pq_1,j}(X) = W_{pq_1}^T \otimes X^{j:j+k_{pq_1}-1},$$

$$H_{pq_1} = \sum_{j=1}^{n} tanh(\mathsf{C}_{pq_1,j}(X)),$$

$$H_{pq_2} = \sum_{j=1}^{n} \mathsf{C}_{pq_2,j}(H_{pq_1}),$$

$$H_{pq_3} = \sum_{j=1}^{n} \mathsf{C}_{pq_3,j}(X),$$

$$H_{pq} = tanh(H_{pq_2} + H_{pq_3}),$$

where + represents the element-wise addition and H_{pq} is the final output from the q^{th} residual block that used the initial input matrix from the output of p^{th} multi-filter convolutional block. X is the input matrix to each of the

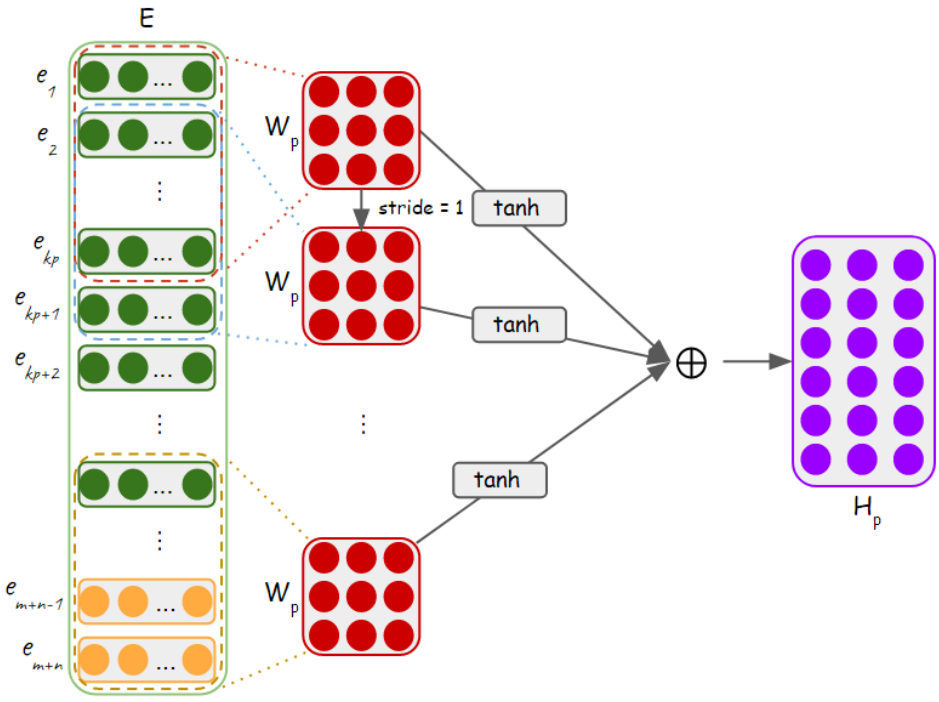


Figure 19: A general architectural overview of 1-D convolution with stride 1.

residual blocks. The first residual block is fed with the output of the multi-filter convolution layer. Finally, the output of each of the final residual blocks is concatenated together to use in the next step. The final output can be formulated as:

$$H = \sum_{1}^{p} H_{pq}$$

where p is the total no of filters used in the multi-filter convolution layer.

Attention Layer

The final output matrix H is typically reduced to a vector using the max-pooling operation before passing it to a classifier. However, in this model, we used an additional label attention step as suggested by Mullenbach et al. [247]. The idea is that some words have higher weights for a label for multi-class classification. Therefore, the label attention can select the most relevant k-grams from the text that can benefit in predicting the correct label. Formally, the procedure is to create a vector parameter U for the labels and then compute the matrix-vector product HU. Then we use a softmax layer to obtain the word distribution in the text.

$$\alpha = softmax(HU)$$

where α is the attention vector. To get the final vector representation from the attention layer we again perform a matrix multiplication between the attention vector α and the input matrix H. The final output is formulated as

$$V = \alpha^T H$$

Output Layer

The output layer is a superficial linear layer that takes the input V from the attention layer. The score vector of all the labels is obtained using the sum-pooling operation on the output vector resulting from a linear transformation. The final probability vector is calculated using sigmoid activation on the score vector for multi-class classification, such that $Y = VW$, where W of dimension $((p \times d^{pq}), l)$ is the weight matrix. Here, p is the total number of convolution filters used in the multi-filter convolution step, and d^{pq} is the output dimension from the residual convolution layer. l is the output dimension, the total number of labels that we are classifying. The score vector $\hat{Y}$ can be formulated as :

$$\hat{Y} = pooling(\sum_{j=1}^{l} Y_{ij})$$

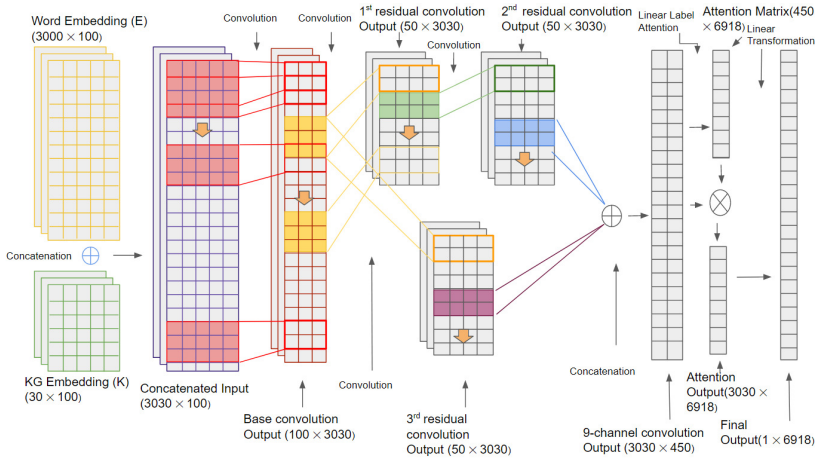


Figure 20: Full implemented architecture of "KG-MultiResCNN."

and the final predicted vector is :

$$\tilde{Y} = \sigma(\hat{Y})$$

6.4 Results

In this section, we evaluate the effectiveness of the *KG-MultiResCNN* against the baseline state-of-the-art approaches. To reproduce the results and further improvements, we made the implementation of the *KG-MultiResCNN* publicly available [40] and the details of the architecture are illustrated in Figure. 20.

We conducted several experiments with different parameters to determine the optimal operation settings for our model. We found that using 100-dimensional embedding vectors for the input word embedding yielded better performance than using higher-dimensional embedding vectors. Additionally, the number of words in the clinical text played a significant role in the model's performance, with a maximum of 3000 words resulting in the best performance. We also discovered that using a maximum of 30 medical entities extracted from the clinical text led to optimal performance, and architecture with nine CNN channels was the most advantageous for modelling this number of words. For the combined input of word embeddings and KG embeddings, the model performed best with two residual layers. Although the complexity of the "KG-MultiResCNN" model was relatively high, it had comparable computational costs to the "MultiResCNN" model. However, if the number of words and extracted medical entities is higher, more CNN channels and/or residual layers would be needed, leading to increased computational costs.

[40] `https://KG-MultiResCNN.ai-research.net`

Dataset

Medical Information Mart for Intensive Care (MIMIC-III) [177] is one of the largest labelled datasets of clinical texts with clinical records of around 40 thousand patients. Also, it is used by most of the state-of-the-art approaches [212, 247, 133, 373, 386]. Therefore, MIMIC-III is adopted in this work to be the evaluation dataset. Similarly to Mullenbach et al. [247] and Li and Yu [212], we use in this work the "Discharge summaries" which contain a general description of the patient, starting from their medical history to the final discharge notes. On top of that, we aim also to assess the capability of *KG-MultiResCNN* on predicting the ICD codes from the *clinical descriptive texts* and without using the discharge notes. We mean by *clinical descriptive texts*, texts that describe the clinical case (e.g. lab tests and clinical observations) without any explicit or implicit clue of the diagnosis and they include clinical notes, nursery observations and free-text notes from medical examinations such as radiology, electrocardiography, echocardiography, and respiratory check examinations. Following the baseline approaches (e.g. [212]), we consider two experiments, one will full codes (4216) and the second one with the top occurring 50-codes. This means that only clinical instances, that are assigned to at least one of the top 50 most frequent codes, are considered. This is because most of the ICD codes are assigned to very few hospital admissions.

Evaluation metrics

KG-MultiResCNN is a multi-class classifier, distinguishing between several ICD codes. It is customary to evaluate this kind of classifier at a range of thresholds $p_\tau \in [0; 1]$ for the decision $p > p_\tau$ and then represent the results in the form of Receiver Operating Characteristic (ROC) curves and Area Under ROC (AUROC). However, although the distinction is important, it may not properly address clinical usefulness [262, 235, 334, 338, 352, 327]. More specifically, a false negative prediction is more harmful than a false positive decision. In that case, a model with high sensitivity may be preferable to a model with high specificity and low sensitivity. In other words, a model is clinically useful if its decisions for patients lead to a better ratio between benefits and harms compared to not using the model. Therefore, we employed other evaluation metrics: AUC, Precision@5 (P@5), Precision@8 (P@8), and Precision@15 (P@15). Since the classes (ICD codes) are not supposed to be balanced, micro and macro averaging are adopted for better computation of the average score among the different classes.

Baselines

Because the main contribution of *KG-MultiResCNN* is enhancing *MultiResCNN* [212] with external knowledge guidance, the main comparison is against *Mul-*

tiResCNN. In addition, we consider the following baselines:

- **Logistic Regression** (LR): Mullenbach et al. [247] used Logistic Regression (LR) to predict ICD codes using a unigram bag-of-words vector for all words in the MIMIC-III text data.
- **SVM**: Perotte et al. [279] experimented with hierarchical and flat ICD code prediction on MIMIC-II using Support Vector Machine (SVM). Later, Xie et al. [371] used also SVM for hierarchical ICD code prediction on the MIMIC-III dataset. Their model performed moderately with 10,000 unigram word vectors and with TF-IDF weighting.
- **CNN**: Mullenbach et al. [247] experimented with the performance of 1D-CNN on classifying ICD codes from MIMIC-III clinical notes.
- **Bi-GRU**: Mullenbach et al. [247] achieved modest performance by applying the Bi-GRU [76] for ICD classification with MIMIC-III clinical notes.
- **C-LSTM-Att**: Shi et al. [315] used an LSTM based language model called the Character-aware LSTM-based Attention (C-LSTM-Att). The model used an attention mechanism to handle the mismatch between notes and ICD codes and was used to predict the top 50 ICD codes from the MIMIC-III dataset.
- **LEAM**: Wang et al. [354] proposed a text classification model called the Label Embedding Attentive Model (LEAM) that predicts the top 50 ICD codes from the MIMIC-III dataset. The model projects the embedding of words and labels in the same latent vector space and calculates the similarities between the embeddings.
- **CAML**: Mullenbach et al. [247] introduced the Convolutional Attention Network for Multi-Label classification applied on ICD code classification using MIMIC-III notes. The model achieved high performance for multi-label ICD code classification.
- **DR-CAML**: As an extension of CAML, Mullenbach et al. [247] introduced the Description Regularized CAML. The model used the text description of the codes for better prediction accuracy.

Comparison against the baselines

In this comparison, only "Discharge summary" is considered because it is the only type of note used by the baselines. As the main comparison, we compared KG-MultiResCNN against all baseline approaches mentioned above. Table 26 presents the comparative results in terms of Micro and Macro F1-score averages for both "full-codes" and "50 codes" experiments. It is evident from the results that KG-MultiResCNN significantly outperforms all the

baseline approaches, including current state-of-the-art MultiResCNN. Even with the full diagnosis and procedural ICD coding setting, KG-MultiResCNN acquired a Micro F1-score average of 56.1%, surpassing all approaches.

Model	Full Codes		Top-50 Codes	
	Micro(%)		Macro(%)	
LR	27.2	1.1	53.3	47.7
Flat SVM	39.7	-	-	-
Hierarchy SVM	44.1	-	-	-
C-LSTM-Att	-	-	53.2	-
CNN	41.9	4.2	62.5	57.6
Bi-GRU	41.7	3.8	54.9	48.4
LEAM	-	-	61.9	54.0
CAML	53.9	8.8	61.4	53.2
DR-CAML	52.9	8.6	63.3	57.6
MultiResCNN	55.2	8.5	67.0	60.6
KG-MultiResCNN	**56.1** ±0.1	**10.2** ± 0.1	**69.5** ± 0.1	**64.5** ±0.1

Table 26: Comparison results of KG-MultiResCNN against the baseline methods on predicting ICD codes using "Discharge summary" notes in terms of F1-Score. ± indicates standard deviations.

For further evaluation, we compare KG-MultiResCNN against MultiResCNN in terms of predicting the diagnosis ICD codes using the "Discharge summary" notes. Table 27 shows the comparison results between the two approaches, demonstrating that KG-MultiResCNN achieved better macro and micro F1-score compared to MultiResCNN. It is important to note that the results of MultiResCNN can be slightly different than what was mentioned on the paper [212] as we reproduced them to guarantee a fair comparison. When applied to the "full code" dataset, the guidance of the knowledge graph in KG-MultiResCNN improved the Micro F1-score average by 0.9%. In terms of Macro F1-score average, KG-MultiResCNN is better with 1.7%. Similarly, for the "50-code" dataset, "KG-MultiResCNN" achieved better results compared to MultiResCNN, where the Micro F1-score and Macro F1-score are improved with 1.46% and 3.9%, respectively. The results also show a stable standard deviation for both the "full-codes" and "50 codes"

experiments. Despite the result improvement is marginal, it clearly answers the research question raised in this work and proves that guiding the model with medical knowledge graph embeddings of clinical entities is beneficial in automatic ICD coding.

Results on different note types

Since all the baseline approaches used only "discharge summary" notes which might explicitly comprise the disease, we aim to evaluate the performance of KG-MultiResCNN on the other note types that definitely do not contain an explicit indication of the disease.

Table 28 illustrates a comparative results of "KG-MultiResCNN" with different notes combination for the full code prediction and for top 50 code prediction settings. As anticipated, the model performed better when using only "Discharge summary" notes. By including "Physician" and "Nursing" notes, the results drop slightly, which can be explained by the high dimensionality of the input layer and the complex relationships between the huge number of entities in the text. We assume that a more sophisticated architecture with more layers would work better with a large number of tokens/entities. Another reason could be the huge amount of indirect or irrelevant information that misleads the model. Due to the same reasons, the performance drops significantly when using only "Physician" and "Nursing" notes. However, the results are still promising for using the model in other tasks (e.g. preliminary diagnosis) and/or for further improvements of the model.

Performance

Table 29 presents the performance comparison between *KG-MultiResCNN* and the state-of-the-art baseline *MultiResCNN* from different aspects. As shown in the table, *KG-MultiResCNN* converges after 15 epochs only, whereas *MultiResCNN* took 26 epochs to converge. Also, both models have the same number of training parameters. However, *KG-MultiResCNN* takes about 2185 seconds for each epoch whereas, *MultiResCNN* takes about half of the time. This is due to the higher complexity of *KG-MultiResCNN*. For instance, *KG-MultiResCNN* uses nine convolution channels compared to *MultiResCNN* which uses only six.

6.5 Conclusion

In this study, we presented KG-MultiResCNN, a multi-channel convolutional network model for predicting multi-label ICD codes using clinical text embeddings. KG-MultiResCNN incorporates medical knowledge graph embeddings that capture the relationships between medical entities in the clinical

Model	Full Codes				Top-50 Codes				P@8	P@15	P@5
	Micro(%)		Macro(%)		Micro(%)		Macro(%)				
	F1	AUC	F1	AUC	F1	AUC	F1	AUC			
MultiResCNN	55.2	**98.6**	8.5	**90.5**	67.0	94.5	60.6	92.5	59.1	43.7	57.5
KG-MultiResCNN	**56.1** ±0.1	98.4	**10.2** ± 0.1	87.1	**69.5** ±0.1	**94.5**	**64.5** ± 0.1	**92.7**	**59.9**	**44.0**	**57.8**

Table 27: Comparison results of KG-MultiResCNN against KG-MultiResCNN on diagnosis ICD code with "Discharge summary" notes. ± indicates standard deviations.

Notes Type	Full Codes				Top-50 Codes				P@8	P@15	P@5
	Micro(%)		Macro(%)		Micro(%)		Macro(%)				
	F1	AUC	F1	AUC	F1	AUC	F1	AUC			
"Discharge summary" notes	**53.8**	98.4	**10.2**	87.1	**69.06**	94.5	**64.21**	92.7	59.9	44.0	57.8
"Discharge summary" +"Nursing" +"Physician" notes	53.4	98.1	8.8	85.3	68.19	93.7	61.83	91.5	58.9	43.3	57.4
"Nursing" +"Physician" notes	30.5	93.2	2.46	71.5	48.32	82.7	38.13	77.2	42.2	30.7	46.8

Table 28: Comparison results of KG-MultiResCNN on full and top 50 diagnosis ICD codes with multiple note combinations.

	MultiResCNN	*KG-MultiResCNN*
Trainable Parameters (million)	11.9	11.9
Training Time (seconds/epoch)	1026	2185
No of epochs	26	15

Table 29: Performance comparison between *KG-MultiResCNN* and *MultiResCNN*

text. It also considers the relevance of each word by weighting its embedding with a TF-IDF score based on its occurrence in the document and corpus. Results demonstrate that KG-MultiResCNN outperforms state-of-the-art methods, especially with discharge summary notes, which provide critical patient information.

Future research will focus on constructing a medical-specific knowledge graph to address the limitations of the currently adopted knowledge graph, which contains irrelevant relationships. This new graph will be automatically generated from unstructured medical sources like Wikipedia articles and scientific papers. We also plan to combine knowledge representation (via a knowledge graph) with concept representation (via an ontology) to create a model capable of understanding data at three levels: examples from training data, knowledge from the knowledge graph, and the general framework of the data domain.

7 Paper 8: LLM for ICD Coding

Large Language Model in Medical Informatics: Direct Classification and Enhanced Text Representations for Automatic ICD Coding

Zeyd Boukhers(✉), *AmeerAli Khan, Qusai Ramadan, Cong Yang*

(DOI: 10.1109/BIBM62325.2024.10822419)

Abstract Addressing the complexity of accurately classifying International Classification of Diseases (ICD) codes from medical discharge summaries poses a significant challenge due to the delicate and complex nature of medical documentation. This paper investigates the application of Large Language Models (LLM), specifically LLAMA, in enhancing ICD code classification through two distinct methodologies: direct application as a classifier and as a generator of enriched text representations within a Multi-Filter Residual Convolutional Neural Network (MultiResCNN) framework. We evaluate these methodologies by comparing them against state-of-the-art approaches. The investigation reveals the potential of leveraging LLMs to significantly improve classification outcomes by providing deep, contextual insights into the medical texts, thereby facilitating a more precise and effective identification of ICD codes. The versatility of LLAMA in both direct classification and as a preprocessing step for another architecture assesses the model's adaptability and efficacy in addressing the challenges of medical text classification from two perspectives.

Keywords: *ICD Coding, Clinical Text Analysis, Large language Model*

7.1 Introduction

In the last decade, advancements in Deep Learning (DL) and Natural Language Processing (NLP) have transformed healthcare research, driven by the exponential growth in available health data [366, 258, 255, 229]. These advancements have achieved remarkable success in interpreting medical images and processing Electronic Health Records (EHR), particularly through the application of Deep Neural Networks [212, 354, 247]. A critical application of these technologies is in the automatic assignment of International Classification of Diseases (ICD) codes, which is essential for documentation and healthcare administration worldwide [45, 251].

The automation of ICD coding has been a focus of research for more than two decades. Initially, the task relied on manually-created features and gradually evolved to incorporate sophisticated machine learning techniques [308, 279, 315, 212].

Despite these technological advancements, the task of mapping unstructured medical texts to specific ICD codes remains a formidable challenge, primarily due to the intricate and varied nature of medical language and the complex conditions described therein. Consequently, existing models often fall short of capturing the full contextual and semantic depth necessary for accurate ICD coding. Recently, the introduction of Generative LLMs, such as the LLAMA-2 (7b) model, has started to redefine the standards of accuracy and efficiency in text analysis in general. These models leverage unparalleled linguistic understanding capabilities to enhance the correlation between vast documents and entitites [67].

This paper aims to bridge these gaps by employing LLAMA-2 not only as a powerful tool for direct classification but also for generating enriched text representations to be processed by another classifier. These applications are designed to fully exploit the semantic capabilities of LLAMA-2 (7b) to enhance its effectiveness in the domain of medical text interpretation and ICD code classification.

Our main contributions are:

- Adapting LLAMA-2 (7b) for the direct classification of ICD codes and evaluating its effectiveness beyond traditional generative applications.
- Employing raw LLAMA-2 (7b) to create enriched text representations to be processed by Multi-Filter Residual Convolutional Neural Networks (MultiResCNN) for refined classification.
- Evaluating these methodologies on the MIMIC-III dataset and comparing its performance against the baselines.

The remainder of this paper is organized as follows: Section 7.2 reviews related work in the application of LLMs and other approaches in medical coding, Section 7.3 describes our methodologies in detail, Section 7.4 discusses the experimental results, and Section 7.5 concludes with the implications of our findings and potential directions for future research.

7.2 Related Work

Assigning ICD codes to EHR documents is complex, error-prone, and costly, which has driven ongoing research into its automation for decades [206]. This section reviews the most relevant ICD coding techniques divided into four distinct categories.

7.2.1 Traditional Machine Learning Techniques

Assigning ICD codes has often relied on traditional machine learning methods, characterized by manually crafted features and established algorithms [308]. Support vector machines (SVM) have been particularly prevalent,

used for classifying both simple and complex ICD code structures in diverse healthcare settings, from patient records to death certificates [279, 193].

Additionally, other approaches like regular expression-based mapping and adaptive data processing have been employed to improve accuracy and efficiency. Zhou et al. [396] applied regular expressions to link diagnoses with ICD codes, and Ferrao et al. [112] utilized structured EHR data with SVMs. Feature engineering also plays a significant role, with Diao et al. [97] using gradient boosting to process large datasets of discharge texts, effectively managing diverse diagnostic categories.

7.2.2 Neural Network-based Techniques

Over the past decade, neural networks have revolutionized ICD coding due to their versatility and efficacy. Character-level LSTMs, Grounded Recurrent Neural Networks (GRUs), and Hierarchical Attention-bidirectional GRUs (HA-GRUs) are some of the architectures employed to improve the alignment between medical texts and ICD codes [315, 32].

Innovations include mixed embedding models that align word and label vectors for optimal predictions [354], and ensemble approaches that combine CNNs, LSTMs, and decision trees to process various data types, enhancing overall accuracy [373]. Notably, the CAML model by Mullenbach et al. [247] uses label attention within a CNN to boost interpretability and precision, which demonstrated significant improvements when applied to the MIMIC datasets. Notably, Li and Yu [212] introduced one of the most recent and sophisticated approaches, the Multi-Filter Residual Convolutional Neural Network (MultiResCNN), which leverages a one-hot encoded label vector and multiple-filter CNN network enhanced with residual layers and label attention mechanisms to predict multiple ICD codes from discharge summary texts. This model demonstrated its effectiveness on the MIMIC-III dataset, showing notable improvements in both MIMIC-Full codes and MIMIC-50 codes.

7.2.3 Knowledge-enhanced Approaches

Integrating external knowledge sources has proven effective in improving ICD code prediction from medical texts [27, 66, 284]. Kumar Chanda et al. [66] used medical term definitions to refine term embeddings, while Bai and Vucetic [28] enhanced the CAML model with a Knowledge Source Integration (KSI) framework that leverages Wikipedia data to focus on rare diseases, showing improved results on the MIMIC-III dataset.

Structured knowledge has also shown benefits. Choi et al. [77] developed GRAM, which combines medical ontology with deep learning via an attention mechanism to enhance the representation of infrequent medical concepts. KAME [230] and Bao et al. [31] use medical ontologies and ICD

descriptions, respectively, to enhance predictions. Du et al. [100] improve token extraction and the modeling of diagnosis codes' interactions using a graph-based approach.

Boukhers et al. [54] introduced KG-MultiResCNN, a model that combines training data with knowledge from the Wikidata Knowledge Graph to better capture relationships between medical entities. Their model outperformed baseline approaches on the MIMIC-III dataset, demonstrating the effectiveness of using structured external knowledge to enhance deep learning models for ICD coding.

7.2.4 LLM-based Approaches

The application of LLMs for ICD coding has been a focal point of recent advancements in medical informatics. Encouraged by the transformative impact of pretrained Transformer models across various natural language processing (NLP) tasks, their integration into automated ICD coding has garnered significant interest. Silvestri et al. [318] highlight in their work the potential of transformers in a multilingual context to effectively leverage cross-lingual capabilities to enhance ICD-10 coding accuracy across different languages.

In this direction, Biswas et al. [40] introduced TransICD, a transformer-based architecture that employs a code-wise attention mechanism to capture the interdependence among tokens within a document. In addition to enhancing the model's ability to learn code-specific representations, this method overcomes the imbalance in code frequency within clinical datasets through a label distribution aware margin (LDAM) loss function. Liu et al. [220, 219] proposed a Transformer-based model, XR-LAT, to navigate the complexities of automated ICD coding by employing a recursively trained model chain on a predefined hierarchical code tree. This is further complemented by label-wise attention and knowledge-transferring mechanisms, which have shown a significant improvement on macro-AUC on MIMIC-III and MIMIC-II datasets.

7.3 LLAMA for ICD Coding

Our research addresses the complex task of ICD code classification by utilizing the advanced capabilities of the Large Language Model LLAMA-2 [344]. We adapt this model for two critical applications: direct classification of ICD codes and the generation of enriched text representations. This section elaborates on the mathematical foundations and strategic implementations of these methodologies, detailing how they leverage the LLAMA-2 model to meet the specific challenges of medical text analysis.

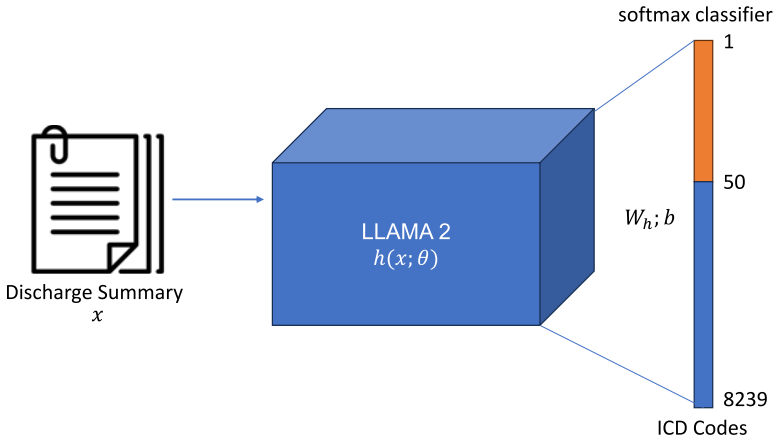


Figure 21: Overview of the LLAMA-2 Classifier Pipeline: the flow from inputting discharge summaries to ICD code classification via the LLAMA-2 model, which processes text data into feature vectors that are classified using a softmax classifier.

7.3.1 LLAMA as Classifier

The $llama\_classifier$ configuration employs a fine-tuning strategy on the LLAMA-2 model to be used as a strong sequence classifier. As depicted in Figure 21, this process involves the integration of a specialized classification head, which is designed to navigate the discrete output space of the ICD codes, that range from 50 top codes to 8239 full codes. The classification is computed as follows:

$$P(y|x;\theta) = \text{softmax}(W_h h(x;\theta) + b) \tag{36}$$

where x denotes the input text encapsulating discharge summaries, and y represents the predicted ICD code. The parameters of the LLAMA model are symbolized by θ, with $h(x;\theta)$ indicating the final hidden representation extracted by the model. W_h and b stand for the weights and bias of the classification layer, respectively. This configuration effectively captures the complex semantics embedded in medical texts.

The objective of the optimization in the $llama\_classifier$ configuration is to minimize the cross-entropy loss between the predicted and actual ICD codes across all training samples. This is formally defined as minimizing the following loss function:

$$L(\theta) = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log P(y = c|x_i;\theta), \tag{37}$$

where $y_{i,c}$ is the ground truth label, and $P(y = c|x_i;\theta)$ is the probability of class c predicted by the model for the input x_i parameterized by θ.

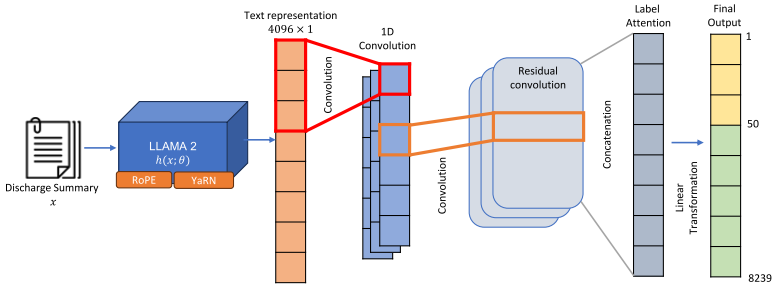


Figure 22: Architecture of the LLAMA-2 Model for Text Representation: the flow from inputting discharge summaries to the final classification of ICD codes. The LLAMA-2 model utilizes RoPE and YaRN techniques within its architecture to enhance text representation, which is then processed through successive convolutional and residual layers. The process culminates in a label attention mechanism that outputs predictions across ICD classes.

7.3.2 LLAMA as text representation

Beyond direct classification, we employ LLAMA-2 (7b) to generate enriched text representations that provide a deeper semantic understanding of medical texts. This process transforms discharge summaries into high-dimensional vectors that capture essential clinical information, which is then fed to the classifier. To this end, we employed the raw LLAMA-2 model without further fine-tuning. The objective is to find the informativeness level of the generated embeddings relative to the clinical content they represent. This is quantitatively measured as the discrepancy between the embeddings and their idealized representations that would yield perfect classification accuracy given any classification model or architecture γ. The overall overview of this process is illustrated in Fig 22. The MultiResCNN architecture is detailed in [212, 54]

Generating Text Representations Within this configuration, the LLAMA-2 model is utilized primarily to generate enriched text representations of discharge summaries through an embedding process that is optimized for capturing the contextually relevant features necessary for precise ICD coding. Due to the length of clinical text, the model's context window is expanded from the standard 4096 tokens to a broader capacity to enable it to fully encompass lengthy discharge summaries. To this end, we applied the Rotatory Position Embedding (RoPE) technique [329], which is further refined by the 'Yet another RoPE extension method' (YaRN) [273]. These techniques adaptively adjust the context size to improve the model handling of extensive texts. The RoPE technique mathematically rotates the positional encodings based on their sequence position to enhance the model's ability to maintain positional context over longer sequences. The YaRN method extends this by

dynamically scaling these rotations according to the document length, and adapting the model to the specific demands of extensive medical texts. The expansion of the context size is modelled as follows:

$$S_{new} = S_{original} \cdot \alpha \tag{38}$$

where S_{new} represents the expanded context size, $S_{original}$ is the initial context size, and α is the scaling factor, dete the positional encoding function of the transformer-based LLAMA-2 model to include a dynamic adjustment parameter β, influenced by YaRN. The modified positional encoding function is then:

$$PE_{pos} = \sin(pos \cdot \beta), \cos(pos \cdot \beta) \tag{39}$$

where *pos* is the position index and β is a factor that adjusts the influence of the position on the embedding, which reflects the Rotatory Position Embedding enhancements.

$$\beta = \log(\alpha) \cdot \rho \tag{40}$$

ρ is an adjustment coefficient determined during the training phase, and $\log(\alpha)$ represents the logarithmic scaling introduced by YaRN to modulate the embedding dynamics based on the extended context window. They collectively ensure that the model adaptively adjusts to the demands of larger textual inputs, which then improves comprehension and classification accuracy of detailed medical documents.

MultiResCNN Classifier To map the clinical text representation to the ICD codes, we followed the work of Li and Yu [212] by building a multi-filter 1-dimensional Convolutional Neural Network architecture. Feeding the high-dimensional vector space of 4096 dimensions to the MultiResCNN causes computational and temporal challenges posed by such high dimensionality. Therefore, we explore two primary strategies for dimensionality reduction: employing convolutional neural network (CNN) layers before input into the MultiResCNN and integrating multiple residual layers within MultiResCNN itself for efficient dimension compression. Both strategies collectively address the dual objectives of reducing computational load and optimizing classification performance. Incorporating CNN layers before MultiResCNN effectively compresses the high-dimensional vectors, relaxes computational requirements and enhances feature extraction for classification. The dimensionality reduction process is then described mathematically by:

$$V_{reduced} = \text{Conv}(V_{4096}; \theta_{conv}) \tag{41}$$

Here, V_{4096} is the initial dense text representation, and $\text{Conv}(\cdot)$ represents the convolution operation, parameterized by θ_{conv}. The resultant $V_{reduced}$

vector offers a dimensionally optimized form, streamlined for efficient classification. This allows to condensing of complex medical narratives into a more manageable and informatively rich format, specifically optimizing the data for subsequent classification tasks by MultiResCNN.

Concurrently, stacking multiple residual layers within MultiResCNN allows for adaptive, in-model dimensionality reduction using a deeper network. This effectively allows the model to learn more complex patterns from the data without losing important information from earlier layers. This process reduces the number of dimensions within the model itself, as opposed to using CNN blocks outside the model. Specifically, a multi-layer residual mechanism is incorporated to compress the representation to 128 dimensions. This allows the model to balance computational efficiency with the capacity to learn complex data representations:

$$V_{final} = \sum_{k=1}^{K} \text{ReLU}(\text{ResLayer}_k(V_{reduced}; \theta_k)), \tag{42}$$

where $ResLayer_k(.; \theta_k)$ represents the k-th residual layer operation, and $ReLU(.)$ is the activation function that introduces non-linearity, enhancing the model's ability to model complex relationships.

7.4 Experimental Results

In this section, we evaluate the effectiveness of both applications of LLAMA against baseline approaches. The codes of all the experiments can be found in the GitLab repository[41]

7.4.1 Dataset

The Medical Information Mart for Intensive Care III (MIMIC-III) database [177] offers a comprehensive collection of de-identified clinical records for approximately 40,000 patients. In line with our preceding research [54] and that of Li and Yu [212], this study utilizes "Discharge Summaries" from MIMIC-III. These summaries provide detailed overviews of patient care, from initial medical history. Consistent with established benchmarks [212], for both applications of LLAMA, we conduct two sets of experiments: one encompassing the full range of 4216 ICD codes and another focusing on the 50 most frequently occurring codes. For the latter, we exclusively analyze clinical instances that correspond to at least one of these dominant codes. Diverging from prior methodologies, our preprocessing phase preserves textual integrity by maintaining case and punctuation, thereby safeguarding the semantic depth essential for precise understanding and classification.

[41] `https://gitlab.com/fdda1/automatic-diagnosis-from-clinical-texts/-/tree/main/icd_coding`

7.4.2 Evaluation Metrics

When assigning ICD codes, missing a true code (false negatives) is often more detrimental than incorrectly detecting a code that is not present (false positives). Therefore, a model that tends toward reducing false negatives is often considered more beneficial, as it minimizes the risk of overlooking critical diagnoses. To capture this, the evaluation metrics include Area Under the Receiver Operating Characteristic Curve (AUC), F1 Score, and Precision at various thresholds; P@5, P@8, and P@15. Given the likely imbalance across ICD codes, we apply both micro and macro averaging methods to compute a more representative average score across the various classes.

7.4.3 Baselines

To evaluate the performance of LLAMA-based methodologies for ICD coding, we compare the obtained results primarily with those achieved by MultiResCNN [212], as well as with the results of other benchmark approaches:

- **C-MemNN**: Prakash et al. [284] introduced the Condensed Memory Neural Network (C-MemNN) that combines a memory network with an iterative condensed memory mechanism. This model was notable for its performance on the MIMIC-III top-50 code dataset.
- **C-LSTM-Att**: Shi et al. [315] presented the Character-aware LSTM-based Attention (C-LSTM-Att), an LSTM model enhanced with an attention mechanism to address the alignment between clinical notes and ICD codes, targeting the top 50 codes in the MIMIC-III dataset.
- **CAML**: The Convolutional Attention network for Multi-Label classification (CAML), introduced by Mullenbach et al. [247], demonstrated strong capabilities in multi-label ICD code classification from MIMIC-III notes.
- **DR-CAML**: An evolution of CAML, also by Mullenbach et al. [247], the Description Regularized CAML (DR-CAML) leverages codes' descriptions to enhance prediction accuracy.
- **KG-MultiResCNN**: In our previous research [54], we augmented the MultiResCNN with knowledge graph embeddings from Wikidata to provide an external knowledge dimension to the classification task.
- **XR-LAT-BootstrapHyperC**: Liu et al. [220, 219] initialize sibling node weights uniformly and employ hyperbolic space for dynamic correction to capture hierarchical relationships within ICD codes. Unfortunately, the results are reported for full code configuration only.

- **TransICD**: Biswas et al. [40] propose to use a transformer encoder for token representation and then a self-attention mechanism to classify the labels. Unfortunately, the results are reported for the top 50 codes configuration only.

All comparisons rigorously adhere to the experimental protocol established by MultiResCNN [212]. Due to varying experimental protocols, many other relevant studies were excluded from the direct comparison, as their methodologies could not be precisely replicated or their results reliably reproduced.

7.4.4 Results

For our empirical analysis, we conducted a series of experiments on the MIMIC-III dataset [177], focusing initially on the top-50 ICD codes to fine-tune our model configurations. After identifying the optimal settings, we applied them to the full-code dataset. The hyperparameters for fine-tuning the LLAMA-based classifier were determined empirically through rigorous experimentation, to ensure robust performance across different model settings. These parameters are detailed in Table 30. Similarly, the selection of hyperparameters for the MultiResCNN classifier, presented in Table 31, was also empirically derived to achieve the best possible outcomes in our testing scenarios.

Hyperparameter	**Value**
Number of Training Epochs (Top-50 codes)	5
Number of Training Epochs (Full codes)	3
Learning Rate Scheduler Type	Cosine
Warmup Ratio	0.03
Maximum Gradient Norm	0.3
Learning Rate	1e-4
Use BF16 Precision	True
Gradient Checkpointing	True
Gradient Accumulation Steps	32
Per-device Training Batch Size	2
Per-device Evaluation Batch Size	2
Adam Beta2	0.99
Weight Decay	0.0

Table 30: The hyperparameters used in fine-tuning LLAMA classifier

As mentioned in Section 7.3, the high-dimensionality of text representation is handelled via two strategies: CNN Reduction and Residual Layer.

For dimension reduction via CNN layers before MultiResCNN integration, we experiment with dimensions of length: 1024, 786, 512, 300 and 100

Hyperparameter	Value
Number of Training Epochs (Top-50 codes)	15
Number of Training Epochs (Full codes)	25
Patience level	3
Learning Rate	1e-4
Per-device Training Batch Size	2
Per-device Evaluation Batch Size	2

Table 31: The hyperparameters used in fine-tuning MultiResCNN classifier

denoted as **CNN-1024**, **CNN-768**, **CNN-512**, **CNN-300**, and **CNN-100**, respectively. These reductions significantly decrease the model's complexity to 118.73M, 82.42M, 68.85M, 41.95M, and 14.83M trainable parameters, respectively. Among these, the **CNN-1024_map_256** model, which compresses the original 4096 dimensions to 1024 and further maps these to a 256-dimensional output from each residual block, emerges as the most efficient, striking a balance between computational efficiency and model performance. For the multi-layer residual mechanism, we experimented with:

- The **Residual-4096** configuration bypasses dimension reduction, directly incorporating the full 4096-dimensional text representation into MultiResCNN.
- Conversely, **Residual-128** utilizes a multi-layer residual mechanism within MultiResCNN to compress the representation to 128 dimensions, resulting in total trainable parameters of 1292.11 million (M) and 1474.37M, respectively.

Table 32 presents the evaluation metrics for various model configurations trained on 2,000 tokens. A subsequent expansion to 6,000 input tokens reiterates the advantageous pattern of dimensionality reduction, as documented in Table 33. Although **Residual-128** slightly outperforms **CNN-1024_map_256**, the latter maintains optimal performance alongside reduced computational demand. This observation underscores the efficacy of the dimensionality reduction strategy, prompting its application to full-code model training.

To further validate the findings in Table 33, we extended the experiments to compare **CNN-1024_map_256** against the residual layer mechanism on 6000 tokens. The results presented in Table 33 demonstrate that **CNN-1024_map_256** achieves comparable results to the residual layer mechanism. They also reveal that the LLAMA model exhibits enhanced performance with longer input sequences. This finding diverges from the conclusions of [212], which posited limited benefits from extended sequences when employing MultiResCNN in isolation, thereby highlighting the unique

	F1 Macro	F1 Micro	AUC Macro	AUC Micro	P@5
Residual-4096	0.5183	0.6025	0.8722	0.9011	0.5997
Residual-128	0.5305	**0.6117**	**0.8747**	**0.9061**	**0.6024**
CNN-100_map_50	0.4358	0.5280	0.8093	0.8568	0.5342
CNN-300_map_50	0.4473	0.5514	0.8418	0.8772	0.5710
CNN-768_map_50	0.5127	0.5903	0.8504	0.8849	0.5776
CNN-1024_map_256	**0.5320**	0.6114	0.8657	0.8994	0.5991
CNN-512_map_256	0.4662	0.5642	0.8475	0.8851	0.5803

Table 32: *llama_representation + multirescnn* evaluation results trained on only 2k tokens with different convolution blocks

advantage of our combined approach in accommodating and benefiting from extended textual inputs.

Statistical tests comparing the configurations in Tables 32 and 33 demonstrate significant enhancements in model performance as the token counts increase, with p-values indicating robust statistical significance (p-value for F1 Macro = 0.00157; p-value for F1 Micro = 0.0098). These results confirm the substantial improvement in performance metrics with expanded input sequences.

	F1 Macro	F1 Micro	AUC Macro	AUC Micro	P@5
Residual-4096	0.6234	0.6907	**0.9151**	**0.9364**	0.6508
Residual-128	0.6237	0.6739	0.9075	0.9316	0.6459
CNN-1024_map_256	**0.6258**	**0.6912**	0.9138	0.9361	**0.6517**

Table 33: "*llama_representation + multirescnn*" evaluation results trained on 6k tokens

Tables 34 and 35 present the outcomes of our main experiments, compared against the outcomes of the above-listed baselines. The results of these approaches are obtained from the original papers. Within the scope of the top-50 codes, the *llama_classifier* yields satisfactory outcomes, surpassing the *MultiResCNN* in certain metrics such as AUC. This performance underscores LLAMA's proficiency in comprehending complex clinical texts and effectively classifying them across a broad range of categories. The combined configuration of "$llama_representation + multirescnn$" on the other hand, surpasses the baselines, including those relying on external knowledge like DR-CAML and KG-MultiResCNN, across various important metrics: it increases the F1 score by 3.2%, the AUC by 1.2%, and the precision at 5 (P@5) by 1.67%. This combination exploits the LLAMA model's advanced ability to represent the semantics of medical texts, providing a more contextually rich input to the MultiResCNN.

For fullcodes tasks, the *llama_classifier* model significantly underperforms compared to all approaches, with lower scores across all evaluation

	F1 Macro	F1 Micro	AUC Macro	AUC Micro	P@5
C-MemNN [284]	-	-	0.833	-	0.420
C-LSTM-Att [315]	-	0.532	-	0.900	-
CAML [247]	0.532	0.614	0.875	0.909	0.609
MultiResCNN [212]	0.606	0.670	0.899	0.928	0.641
KG-MultiResCNN* [54]	**0.645**	0.691	-	-	-
DR-CAML* [247]	0.576	0.633	0.884	0.916	0.618
TransICD [40]	0.562	0.644	0.894	0.923	0.617
llama_classifier	0.5802	0.6357	0.9011	0.9241	0.6261
"*llama_representation + multirescnn*"	0.6258	**0.6912**	**0.9138**	**0.9361**	**0.6517**

Table 34: Evaluation results for top-50 codes. An asterisk * next to a method denotes the incorporation of external knowledge. The **bolded** result highlights the best performance across all methods, whereas the underlined result signifies the top performance among methods that do not utilize external knowledge.

metrics. This could be due to several factors. First, the fine-tuning process for the LLAMA model utilized a relatively small dataset, which is much smaller than the extensive dataset used during its initial training. This difference in dataset sizes likely constrained the model's ability to effectively adapt to the complexity of full ICD code classification.

Moreover, the fundamental structure of LLAMA, when exclusively utilized as a classifier, may not be suitable for handling label granularity inherent in ICD coding tasks. ICD codes encompass a wide range of specificity and scope, and the model's emphasis may lean towards excessively specific or overly broad codes, negatively impacting its performance. The ability to finely adjust sensitivity to label granularity is essential in such scenarios and may necessitate further model adaptations or alternative approaches better suited for the task.

In contrast, the "*llama_representation + multirescnn*" model achieved comparable results compared to the baselines but with lower accuracy compared to the experiments with 50 codes. Specifically, KG-MultiResCNN significantly outperforms *llama_representation + multirescnn* which indicates that external knowledge can mitigate issues related to data sparsity and label granularity by compensating for the insufficient coverage of training data across all codes. Furthermore, the similar outcomes among models that do not utilize external knowledge suggest that the complexity and sophistication of the model architecture are not decisive factors in addressing data sparsity and label granularity.

In these experiments, the XR-LAT-BootstrapHyperC model achieved the best performance without relying on external knowledge. While the paper does not provide specific details on the model's complexity, it is inferred that XR-LAT-BootstrapHyperC requires substantially longer training times compared to "*llama_representation + multirescnn*" due to the inclusion of multiple training processes. This likely leads to higher computational costs, which could pose limitations for practical deployment, particularly in

environments with restricted computational resources. Additionally, the use of hyperbolic embeddings in XR-LAT-BootstrapHyperC, which are adept at capturing hierarchical relationships inherent in data, is presumed to be more effective than traditional Euclidean embeddings for modelling relationships between entities in the text.

	F1 Macro	F1 Micro	AUC Macro	AUC Micro	P@8	P@15
MultiResCNN [212]	0.085	0.552	0.910	0.986	0.734	0.584
KG-MultiResCNN* [54]	0.102	**0.651**	-	-	-	-
DR-CAML* [247]	0.086	0.529	0.897	0.985	0.690	0.548
CAML [247]	0.088	0.539	0.895	0.986	0.709	0.561
XR-LAT-BootstrapHyperC [220, 219]	**0.108**	0.583	**0.946**	**0.99**	**0.749**	**0.599**
llama_classifier	0.0241	0.3909	0.8604	0.9793	0.6304	0.4789
"*llama_representation + multirescnn*"	0.0688	0.5324	0.8937	0.9838	0.7364	0.5811

Table 35: Evaluation results for full codes. An asterisk * next to a method denotes the incorporation of external knowledge. The **bolded** result highlights the best performance across all methods, whereas the underlined result signifies the top performance among methods that do not utilize external knowledge.

7.5 Conclusion

In this paper, we explored the application of the LLAMA-2 model for ICD code classification. We applied the model in two different configurations: as a direct classifier and as a generator of enhanced text representations. The obtained results show that using LLAMA-2 as a standalone classifier performs less effectively, especially when dealing with a wide range of ICD codes. However, when combined with a Multi-ResCNN classifier, it demonstrates strong capabilities, especially for a smaller set of codes. This configuration takes advantage of the rich semantic information captured by LLAMA-2, improving the classifier's accuracy in classifying medical texts. However, when expanded to a classification with a larger set of codes, the performance aligns more closely with baseline models, underscoring the challenges posed by label granularity and the sparsity of training data.

For future work, we plan to enhance our model's ability to manage the complexities of full ICD code classification by integrating external knowledge in the form of a graph. This approach will aim to compensate for the sparsity of training data by embedding rich, context-specific information directly into the model's training process, potentially through knowledge graphs that map relationships among medical conditions, symptoms, and treatments. Additionally, we propose to develop a more unified architecture that combines the strengths of large language models, with the robust feature extraction capabilities of residual CNNs. This integration will leverage the deep contextual insights of transformers and harness the architectural benefits of residual learning to enhance training efficiency and model performance.

8 Paper 9: Sentiment Analysis for Cryptocurrency Foracasting

Beyond Trading Data: The Hidden Influence of Public Awareness and Interest on Cryptocurrency Volatility

Zeyd Boukhers(✉)*, Azeddine Bouabdallah, Cong Yang, Jan Jürjens*

(DOI: 10.1145/3583780.3614790)

Abstract Since Bitcoin first appeared on the scene in 2009, cryptocurrencies have become a worldwide phenomenon as important decentralized financial assets. Their decentralized nature, however, leads to notable volatility against traditional fiat currencies, making the task of accurately forecasting the crypto-fiat exchange rate complex. This study examines the various independent factors that affect the volatility of the Bitcoin-Dollar exchange rate. To this end, we propose *CoMForE*, a multimodal AdaBoost-LSTM ensemble model, which not only utilizes historical trading data but also incorporates public sentiments from related tweets, public interest demonstrated by search volumes, and blockchain hash-rate data. Our developed model goes a step further by predicting fluctuations in the overall cryptocurrency value distribution, thus increasing its value for investment decision-making. We have subjected this method to extensive testing via comprehensive experiments, thereby validating the importance of multimodal combination over exclusive reliance on trading data. Further experiments show that our method significantly surpasses existing forecasting tools and methodologies, demonstrating a 19.29% improvement. This result underscores the influence of external independent factors on cryptocurrency volatility.

Keywords: *Cryptocurrency Forecasting, Public Awareness, Trading Data, Ensemble Learning, Deep Neural Networks*

8.1 Introduction

Since the mining of the first Bitcoin's genesis block in 2009, cryptocurrencies have redefined the landscape of financial assets. By January 2022, the total market capitalization of major cryptocurrencies nearly reached an astounding $1 trillion[42]. The global cryptocurrency market exhibits a Compound Annual Growth Rate (CAGR) of 30% from 2019 to 2026[43]. This remarkable growth not only offers lucrative investment and trading opportunities

[42]https://gadgets.ndtv.com/cryptocurrency/news/bitcoin-price-btc-cryptocurrency-market-crash-usd-1-trillion-coinmarketcap-2726233

[43]https://www.globenewswire.com/news-release/2021/04/12/2208331/0/en/At-30-CAGR-CryptoCurrency-Market-Cap-Size-Value-Surges-to-Record-5-190-62-Million-by-2026-Says-Facts-Factors.html

but also contributes to further expansion of the cryptocurrency market, fostering a virtuous cycle. In contrast to traditional FIAT currencies, which are regulated by central banks and governing bodies, cryptocurrencies are entirely decentralized. Transactions are validated and processed via a cryptographic network of nodes and are logged on a blockchain – a digital transaction ledger [78]. These unique properties make forecasting the fiat-crypto exchange rate (or simply, cryptocurrency price forecasting) an exceptionally complex and often error-prone task [6]. Given this complexity, financial analysts and AI experts continually dissect the market to deepen their understanding of price fluctuation trends [5, 6, 14, 197, 74, 172, 199].

Recent studies focused on forecasting cryptocurrency prices have predominantly leveraged neural networks due to their exceptional performance in related tasks [199, 224, 253, 283, 381]. As a result, these approaches have achieved superior outcomes compared to traditional machine learning and statistical methods [377, 199]. However, a significant limitation of existing methods is that they consider only a handful of factors influencing the cryptocurrency market. The common practice is to infer a predictive function from available training sets and then evaluate the derived functions based on their generalization capabilities, presuming that price series often display homogeneous nonstationarity [283].

In reality, cryptocurrency volatility is influenced by a multitude of factors, distinct from those affecting foreign exchange rates. Factors such as the cryptocurrency mining process's hash rate and public awareness play a critical role in price volatility [197]. For example, studies have revealed a sensitivity of cryptocurrency prices to public opinion sentiments [5, 204]. Yang et al. [377] proposed that social media sentiment serves as a valuable predictor of future Bitcoin price volatility. Akbiyit et al. [10] have proved this hypothesis in their study. Therefore, a holistic approach that considers these wider influences is essential for understanding the volatility of cryptocurrencies.

To enhance forecasting accuracy, this paper proposes a comprehensive utilization of all available factors that influence cryptocurrency prices. Specifically, we introduce *CoMForE*: ***Co****mprehensivle* ***M****ultimodal Crypto* ***For****ecasting* ***E****nsembler*, an ensemble of multimodal Long Short-Term Memory (LSTM) models for cryptocurrency volatility prediction. This method employs trading data, social media sentiment analysis, blockchain data (including hash rate and network difficulty), and search volumes from search engines. Our focus is on Bitcoin due to its dominant market presence, its popularity among the 7812 existing cryptocurrencies, and the ample data available for analysis. Ultimately, our goal is to facilitate sound investment decision-making through the provision of reliable price forecasts and volatility distribution assessments. The following research questions guide our exploration:

- **RQ1:** Which combinations of data modalities exert the most significant

influence on cryptocurrency volatility?

- **RQ2:** Does the application of ensemble learning to multimodal data yield effective outcomes in the forecasting of cryptocurrency prices?
- **RQ3:** How can we effectively interpret cryptocurrency price forecasting in conjunction with volatility distribution?

To this end, the main contributions of this paper can be summarized as follows:

- Our research encompasses factors influencing cryptocurrency volatility, including trading data, social media sentiments, blockchain metrics, and search engine volumes.
- We introduce an LSTM-based multimodal ensemble learning model that outperforms existing models, delivering reliable decision-making support to investors.
- Beyond price prediction, our model provides fluctuation distribution, enhancing investors' understanding of forecast certainty.
- Comprehensive experiments and analyses affirm the robustness of our approach.
- We provide an open-source implementation of our model for further development and improvement.

8.2 Related Work

In this section, we review the related works divided into three categories:

8.2.1 Traditional Market Price Forecasting

The task of forecasting stock prices has long been a subject of interest for scholars in the realms of finance, statistics [144], and data science [382, 211]. The aim is to empower investors with the tools necessary to augment their profits while mitigating potential losses. One of the early approaches adopted was the application of statistical models. An in-depth comparative study was conducted by Haviluddin et al. [12], examining statistical and machine learning techniques for short-term forecasting using time-series data. The study primarily compared the statistical method ARIMA, Neural Networks, and genetic algorithms. The findings underscored the superior efficiency and reliability of Neural Networks for short-term time series forecasting.

Over the past decade, the advancements in deep neural networks across diverse applications have brought them into the spotlight for financial and economic forecasting, including time-series predictions [157, 312, 184, 259].

Deep learning approaches have demonstrated their ability to significantly outperform other methods, largely due to their capability to learn concealed features of time series and historical market trends. Traditional markets are governed by relatively well-understood factors and rules that directly influence price trends, such as the number of asks, bids, and transactions. This relative clarity facilitates deep learning models in pattern recognition from historical data, leading to highly reliable forecasting, as evidenced by the high accuracy and low error rates observed when applying price, ask, and bid time series alone [259, 312].

However, attempts to apply these cutting-edge deep learning methods—proven effective in traditional markets—to the task of forecasting cryptocurrency prices have not yielded satisfactory results [231]. The distinct characteristics of the cryptocurrency market render it a more complex domain for forecasting.

8.2.2 Machine Learning for Cryptocurrency Price Forecasting

Cryptocurrency market volatility has driven the development of tailored forecasting approaches. Yiying and Yaze [381] proposed LSTM and ANN architectures using price, ask, and bid time series for short and long-term price predictions of Bitcoin, Ethereum, and Ripple. The results showed the LSTM's effectiveness in capturing short-term dynamics.

McNally et al. [236] evaluated LSTM and Bayesian-optimized Recurrent Neural Network models for Bitcoin price forecasting, concluding that the LSTM had marginally superior accuracy. However, they noted the challenge of balancing overfitting and underfitting due to the high volatility of cryptocurrency time series.

Kumar and Rath [199] used only historical trading data for Ethereum price predictions, indicating LSTM's slight outperformance over MLP for short-term forecasts. Meanwhile, Pintelas et al. [283] found LSTM-based and CNN-based models demonstrating almost random walk processes, suggesting the exploration of new approaches.

Chevallier et al. [74] introduced an AdaBoost-based approach for cryptocurrency forecasting that significantly improved performance. Despite its simplicity, AdaBoost outperformed other models like ANNs, LSTMs, KNNs, and SVMs, highlighting its generalizability and interpretability.

8.2.3 Sentiment Analysis and Multimodality for Cryptocurrency Price Forecasting

Krisoufek [197] found that the cryptocurrency market is influenced by various factors, including the number of asks, bids, exchange rates, and notably, public awareness, as confirmed by Akbiyik et al.[10]. Recognizing the challenge of quantifying public awareness, individual sentiments can serve as proxies,

which are believed to affect price trends[309]. Young et al. [189] proposed that cryptocurrency forum sentiments might influence Bitcoin prices. They proposed a model using exclusively sentiment data and found a correlation between price trends and forum sentiments. Leveraging public opinion for finance is not a new concept, and it extends beyond cryptocurrency. Several approaches have been suggested to utilize public sentiment for predicting stock market trends [44, 257, 374]. However, for cryptocurrencies, it tends to have an influence rather than a correlation.

Following these insights, several studies [253, 6, 163] integrated sentiment analysis with LSTM models to predict next-day trading prices. They gathered posts related to cryptocurrencies, assigned each post a sentiment score, and merged these scores with trading data into a singular vector for the LSTM model. Their results revealed a modest enhancement in forecasting outcomes compared to models relying solely on trading data. Furthermore, Huang et al. [163] supported their approach with an autoregressive model that improved accuracy and recall by 18.5%.

However, these studies [253, 6, 163, 74, 283, 199] overlooked the influence of interaction levels on posts' impact such as likes, comments, shares, and exclusively focused on online communities as fluctuation factors, neglecting other crucial factors [197].

In addition to the findings of Krisoufek [197], the other modalities have proven to be correlated with the stock market, such as search volume [82, 333]

8.3 Approach: *CoMForE*

To address our first research question (**RQ1**), this paper proposes a novel strategy for forecasting next-day cryptocurrency volatility by leveraging all indicative factors. To tackle **RQ2**, we introduce an ensemble learning approach that marries adaptive boosting with LSTM architectures, known for their proficiency in learning hidden trends within sequential data, especially for short-term forecasting. As depicted in Figure 23, our approach comprises multiple LSTM weak learners (i.e., forecasters) that collaboratively construct a cryptocurrency volatility prediction model. Each LSTM model j is trained on a subset sampled from the original dataset and subsequently assigned a weighted score w_j based on its performance during the inference phase. Each subset is sampled using the sampling weights $s_{n=1}^{N}$ for the N samples. Here, a sample denotes the input sequence $\mathbf{x}_{t-k}^{t}$ associated with its respective price, where k is the length of the input layer and t is the time stamp (i.e. day). Initially, the sampling weights denoted as $s_{n=1}^{N}$ are set to $\frac{1}{N}$, indicating that all samples carry equal importance in the learning process. Then, for each LSTM forecaster M_j, the following steps are performed:

1. Randomly select $1 < l <= L$ features from the total L features

2. Train the LSTM model M_j using the sampled subset. This involves learning the weights $W_{f,j}$, $W_{i,j}$, $W_{C,j}$, $W_{o,j}$ and biases $b_{f,j}$, $b_{i,j}$, $b_{C,j}$, $b_{o,j}$ of the LSTM.

3. Calculate the errors $e^{(j)}_{n=1:N}$ of the LSTM forecaster M_j for each sample n.

4. Compute the total error E_j of the j-th LSTM forecaster, such that:

$$E_j = \sum_{n=1}^{N} s_n^{(j)} e_n^{(j)} \tag{43}$$

5. Calculate the weight w_j of the j-th forecaster as follows:

$$w_j = \log\left(\frac{1 - E_j}{E_j}\right). \tag{44}$$

6. Update the sampling weights $s_n^{(j)}$ of all samples as follows:

$$s_n^{(j)} = \frac{s_n^{(j)} \exp\left(w_j I\left(y_n \neq M_j(x_n)\right)\right)}{Z_j}, \tag{45}$$

where Z_j is a normalization factor making $s_n^{(j)}$ a distribution.

The normalization factor Z_j ensures that the updated weights $s_n^{(j)}$ form a valid probability distribution. The indicator function $I\left(y_n \neq M_j(x_n)\right)$ checks if the predicted output of the j-th model for the n-th sample does not match the true output for this sample. If true, I returns 1; otherwise, it returns 0.

After all LSTM forecasters are trained, we calculate the final ensemble prediction $\hat{y}_n^{(M)}$ for a given sample n as follows:

$$\hat{y}_n^{(M)} = \frac{\sum_{j=1}^{J} \hat{y}_n^{(j)} w_j}{\sum_{j=1}^{J} w_j}, \tag{46}$$

where J signifies the total number of LSTM forecasters, $\hat{y}_n^{(j)}$ represents the prediction made by the j-th LSTM model M_j for the n-th sample, and w_j denotes the weight of the j-th LSTM forecaster. This weight reflects the relative contribution of each forecaster's prediction towards the final ensemble prediction.

Each LSTM forecaster in our ensemble provides a one-day-ahead forecast of the price of Bitcoin (₿) in US dollars ($). To do this, the forecaster uses a seven-day window of historical data, denoted as $\mathbf{x}_{t-7}^{t}$. Here, $\mathbf{x}_t$ represents a normalized feature vector of length 18 corresponding to day t. This vector combines representations from four distinct modalities: trading data, sentiments, blockchain data, and search volumes. Each of these modalities contributes to the richness and complexity of the feature space and is described in greater detail as follows:

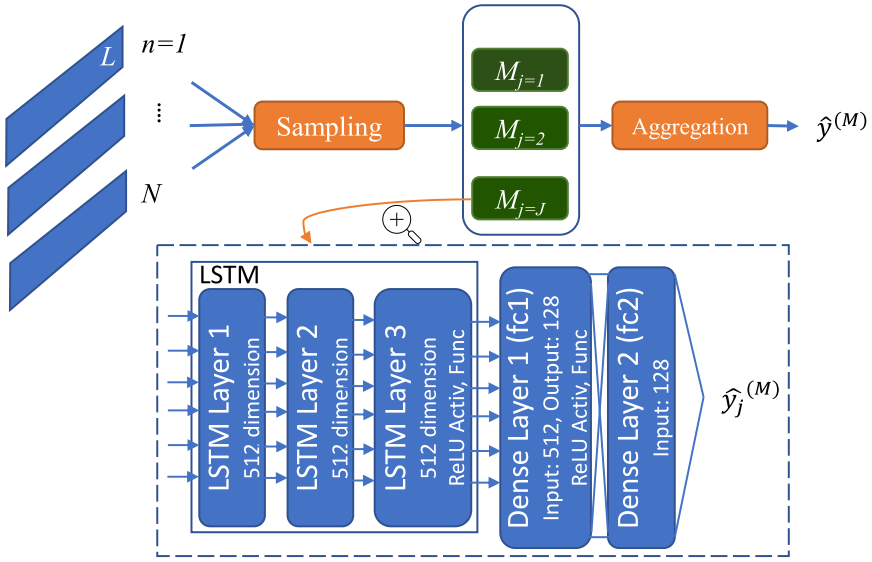


Figure 23: Overview of *CoMForE*'s Architecture for Cryptocurrency Price Prediction

8.3.1 Input modalities

While there is a multitude of cryptocurrencies available, Bitcoin (₿) remains the most prominent, and data related to it is readily accessible. Therefore, this study centres on Bitcoin as the primary use case. We have gathered a comprehensive dataset spanning from 2012 through the end of 2020, comprising the following modalities:

Trading data: Trading data encapsulates a sequential collection of attributes that depict the Bitcoin market's dynamics. Each timestamp t corresponds to a single day, with the opening price recorded at 00:00 and the closing price at 23:59. At each t, the following market characteristics are noted:

- *Open Price ($)*: The price of one Bitcoin (₿) at the beginning of t.
- *High Price ($)*: The peak price of one ₿ reached within the period of t.
- *Low Price ($)*: The lowest price of one ₿ recorded within the period of t.
- *Close Price ($)*: The price of one ₿ at the end of t.
- *Bitcoin Volume (₿)*: The total quantity of ₿ traded during the interval of t.
- *Dollar Volume ($)*: The total value in dollars of ₿ traded during t.
- *Weighted Price ($)*: The weighted average price of ₿ traded during t.

- *Average Transaction Fee ($)*: The average transaction fees charged by the top 20 trading and exchange platforms during t.
- *Number of Transactions*: The cumulative number of transactions carried out during t.

Public awarness: In today's world, tweets serve as an instantaneous source of news. Hence, in this study, they are perceived as a crucial indicator of cryptocurrency price trends. Accordingly, we gathered tweets responding to the wildcard queries: "**Bitcoin**" and "**BTC**", resulting in a corpus of over 120 million tweets. To filter out spam, tweets containing the hashtag "*#Bitcoin*" but not within the main text body were excluded.

The primary goal behind this data collection is to conduct sentiment analysis and incorporate the resulting sentiment scores into our model, thereby encoding the factor of public awareness. To this end, we utilized two widely recognized and publicly available sentiment analysis methodologies: *Vader* [169] and *Deeply Moving*[321]. Both techniques yield a sentiment score θ_γ for tweet γ, ranging from -1 (extremely negative) to 1 (extremely positive), while 0 stands for neutral. The dual-method approach was chosen to balance their inherent sensitivity—intensifying the sentiment score when both methods agree and diminishing it when they disagree.

It is important to note that not all cryptocurrency-related tweets have an equal impact on the market. The key determinant here is the level of engagement (i.e., the tweet's reach and number of interactions). According to recent statistics on Twitter engagement[44], the median number of likes and comments per tweet is zero, implying a substantial portion of tweets receive little to no engagement. Thus, assigning equal sentiment weights to all tweets would not accurately represent the real-world scenario.

To address this, we propose weighting each tweet in a manner that gives higher consideration to those with substantial engagement. Owing to the complexity in determining the relative importance of different engagement factors, we opted for a straightforward approach to compute the weight ω_γ of each tweet γ as the harmonic mean of the number of likes, comments, retweets, and quotes. Subsequently, these weights were normalized using min-max normalization. The final sentiment score θ^*_γ is then multiplied by $\omega\gamma$; $\theta^*_\gamma = \theta\gamma \times \omega_\gamma$.

To create a daily sequence of sentiment scores, all weighted sentiments of tweets posted on a given day t are averaged, generating a singular value representing the overall sentiment on Twitter for that day.

Blockchain details: As asserted in the financial study by Kristoufek [197], blockchain metrics significantly influence the price dynamics of cryptocur-

[44] `https://mention.com/en/reports/twitter/engagement/#2`

rencies, notably ₿. As a result, our strategy incorporates several blockchain metrics that have been demonstrated to impact the market in financial research [197]. Relying on various blockchain data sources[45], we include the following indicators for each day t:

- *Hash Rate:* An estimation of the computational speed at which the ₿ network is operating.
- *Block Size:* The magnitude of a completed ₿ block.
- *Block Time:* The time consumed to mine and generate a new ₿ block.
- *Network Difficulty:* The complexity associated with mining ₿ blocks across the network.
- *Active Addresses:* The aggregate number of functioning addresses.
- *Mining Profitability:* The projected average profit yielded from mining a single ₿ block.

All the above-listed blockchain attributes follow a sequential pattern with a one-day gap between each recorded data point.

Search volumes Public curiosity can often be gauged through a variety of behaviours, and in today's digital age, online searches serve as a significant reflection of societal interest. Thus, we incorporate search volumes into our analysis, specifically focusing on queries involving either the term "*Bitcoin*" or the acronym "*BTC*". Given that Google, being utilized by 52% of the global population[46], is one of the most widely used search engines, we have leveraged the Google API to collect search data for the specified terms. This API returns the volume of searches conducted within a particular time span. These search volumes are then aggregated for each day and normalized, providing an indication of public curiosity on a daily basis.

8.3.2 Fluctuation Analysis

Our primary objective is to augment the decision-making process for cryptocurrency investors. While price prediction is a crucial aspect, it's insufficient given the varying volatility in the cryptocurrency market. During low market fluctuation, model predictions align closely with the actual value. In contrast, during high volatility, the predicted price can significantly diverge. Existing methodologies do not encapsulate this level of fluctuation. Hence, in addressing **RQ3**, we aim to analyze these fluctuations using different variations of the model, each trained using a unique combination of modalities and dropout rates.

[45] Blockchain.com, YCharts, BitInfoChart, and Nasdaq Data Link

[46] `https://review42.com/resources/google-statistics-and-facts/`

Input varieties: Given the model described above, and incorporating data from *Trading*, *Twitter Sentiments*, *Blockchain Details*, and *Online Search Volumes*, we train multiple model variations. Each model variation, utilizing different combinations of modalities, generates a slightly distinct price forecast. In terms of architecture, these model variations only differ in the length of their input layer.

- **Trading Data:** The fundamental input for cryptocurrency price forecasting, consisting of eight essential features: *["Open", "High", "Low", "Close", "Volume BTC", "Volume Currency", "Weighted Price", "Average Fees"].*
- **Twitter Sentiments:** This variant of our model uses only Twitter sentiments, captured through "Weighted Twitter Sentiments" and "Tweet Volumes".
- **Trading Data and Blockchain Details:** This model variant incorporates trading data combined with blockchain details: *["Hash Rate", "Block Size", "Block Time", "Network Difficulty", "Number of Active Addresses", "Mining Profitability"].*
- **Trading Data and Search Volumes:** This model variant merges trading data with *["Online Search Volumes (Google Searches)"]* to improve its predictive power.

Model dropouts: To create variability in our model's predictions and thus estimate an output distribution, we introduce different dropout rates at various layers of the trained model. This method also serves as a regularization technique to counter overfitting. Here are the variants generated with different dropout rates:

- *Models* V_1*,* V_2*,* V_3: These models implement dropout rates of 0.1, 0.2, and 0.35 respectively at the last hidden layer of the LSTM.
- *Models* V_4*,* V_5*,* V_6: These variants use dropout rates of 0.1, 0.2, and 0.35 respectively at the output of the fully connected layer (fc1).

Predicted Price Distribution: We propose that the distribution of fluctuations can provide valuable insights for decision-makers regarding the most suitable actions to take. Paired with the price forecast, they can offer a measure of the forecast's uncertainty. To estimate the parameters (i.e., Mean and Variance) of the distribution based on the sample of outputs, we utilized Maximum Likelihood Estimation (MLE).

Given the outputs $O = o_1, o_2, \cdots, o_{10}$ from the ten model variants, the aim is to estimate the parameter set $\hat{\theta} = \mu, \sigma^2$ that maximizes the likelihood function $L(\theta; O)$. This can be represented as:

$$L(\theta; O) = \prod_{i=1}^{10} \mathcal{N}(o_i; \theta) \tag{47}$$

Then, the set of parameters that maximizes this likelihood function is given by: $\hat{\theta} = \arg\max L(\theta; O)$.

Here, $\mathcal{N}(o_i; \theta)$ denotes the normal distribution of the output o_i with the parameter set θ, and $\hat{\theta}$ represents the estimated parameters that maximize the likelihood function.

8.4 Experiments

In this section, we assess *CoMForE* by comparing its performance to several other baseline approaches. To ensure reproducibility and facilitate further exploration, we have made our implementation available on GitHub[47].

8.4.1 Experimental Setup

Across all experiments, we maintain the same setup and parameters. These are as follows:

- Loss function during training: Mean Squared Error (*MSE*)
- Optimizer: *Adam*, Initial learning rate: *0.0003* and Number of epochs: *200*
- The training process takes place on a GPU server with the following specifications: an AMD Ryzen Threadripper 1950X 16-Core Processor, 128GiB of system memory, and an NVIDIA GV100 GPU.
- All experiments employ the Mean Absolute Error (MAE) as the validation loss, calculated every ten epochs.

8.4.2 Datasets

This study focuses on Bitcoin (₿), the dominant cryptocurrency with readily accessible data. We gathered Bitcoin data from 2016 to 2020, yielding 1825 data points (days). The data was split into 70% for training, 15% for validation, and 15% for testing, as shown in Figure 24. Next, we proceed to a detailed exploration of each data modality.

[47] `https://doi.org/10.5281/zenodo.8265158`

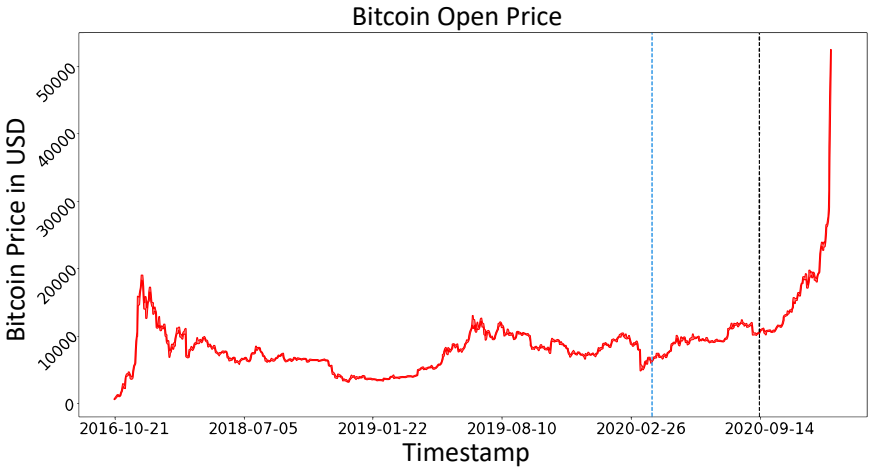


Figure 24: Data split for training, validation, and testing (70% training, 15% validation, 15% testing)

Trading Data: Acquiring comprehensive and accurate historical trading data for cryptocurrencies can be a challenging task due to the presence of missing values and occasional skipped days in many data sources. To address this, we have amalgamated data from several sources, each supplementing the gaps in the other. We also account for the potential variances in values across exchanges, given the lack of a standardized pricing protocol for cryptocurrencies. The sources for our data are as follows:

- **Kaggle:** This open-source dataset comprises 1-minute interval data of ₿ prices, collected from January 1, 2012, to March 1, 2021.
- **Binance:** As a major cryptocurrency exchange, Binance publicly provides its data. In the absence of an API, data was manually extracted from the website.
- **Coinbase:** Unlike Binance, Coinbase offers APIs to gather trading data. As a major cryptocurrency exchange, its data is widely used, including by organizations like Google.

Public Awareness: For sentiment analysis of the collected tweets, a distinct subdiscipline, this paper employs two renowned, pre-trained sentiment analysis tools: *Vader*[169] and *Deeply Moving*[321].

Vader [169], a rule-based model tailored for social media sentiment analysis, has exhibited superior results in recent evaluations compared to various rule-based or machine learning approaches. It is expected to generalize better across different contexts. The model's output helps classify a tweet into one of nine sentiment classes, ranging from extremely negative to extremely positive.

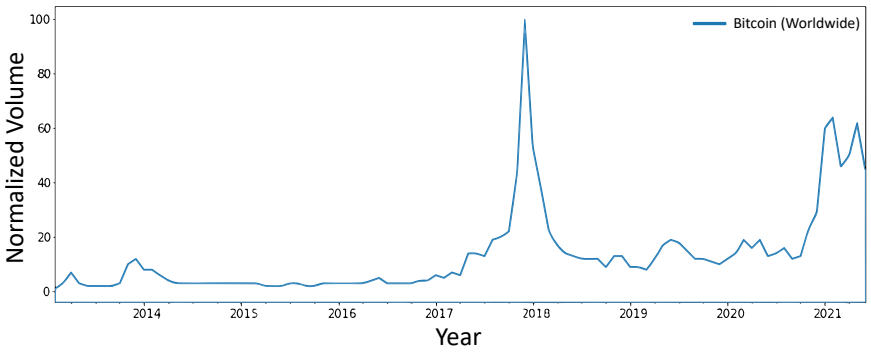


Figure 25: Visualization of *"Bitcoin"* search volume from 2013 to 2021.

Deeply Moving [321] is another tool that examines text as a holistic entity, maintaining word correlations. This method is primarily designed for movie reviews sentiment analysis, meaning that the text is formally represented, which is different from the structure used in Twitter's tweets, such as emojis, hashtags, and abbreviations. Despite the difficulty of fine-tuning this model on tweet sentiments due to the scarcity of labelled data, we aim to utilize both models to harness their unique strengths - Vader's explicit training on tweets and Deeply Moving's ability to comprehend sentiments of an entire sentence.

Both sentiment analysis models (Vader, Deeply Moving) are applied to all previously gathered tweets. The final sentiment score is then derived as the average of the scores from both models.

Blockchain data: As highlighted in Section 8.3.1, the blockchain data has been sourced from various providers[45].

Search volumes: As previously stated in Section 8.3.1, we employed Google's search engine statistics via its API in this study. Figure 25 represents the search volumes associated with the query *"Bitcoin"* from 2013 to 2021.

8.4.3 Baselines

To evaluate *CoMForE*, we selected six notable and modern techniques, each with a unique architecture or different data types, to forecast Bitcoin prices. These include:

- *ARIMA19 [11]:* Utilizes the ARIMA model, focusing exclusively on Bitcoin trading data to forecast the next day's prices.
- *BNN17 [172]:* Employs Bayesian Neural Networks (BNN) to predict future Bitcoin prices, harnessing trading data.

- *LSTM20A [347]:* Adopts a multivariate LSTM model tailored for cryptocurrency price predictions.
- *LSTM20B [246]:* Uses an LSTM architecture, grounded on trading data, to predict price trajectories.
- *GRU20 [102]:* Employs a Gated Recurrent Unit (GRU) infused with recurrent dropout to enhance Bitcoin price prediction accuracy.
- *GRU21 [14]:* While also leveraging a Gated Recurrent Unit, this method has a different predictive focus.
- *AdaBoost21 [74]:* Leverages the traditional AdaBoost algorithm, incorporating multiple decision tree weak learners for its predictions. Its straightforward design achieves noteworthy results, even surpassing more sophisticated methods.

Since the source codes of these approaches are not publically available, we re-implemented them from scratch and reproduce their results. To ensure a fair comparison, we evaluated all the baseline methods, including *CoMForE*, on the exact same dataset within the same time period.

8.4.4 Results and Discussion

Modality Analysis To address **RQ1**, we conducted an evaluation using both *CoMForE* and an *LSTM*-based architecture to investigate the effectiveness of various combinations of data modalities. Both models underwent training for 200 epochs and adhered to the experimental setup outlined earlier. Note that the *LSTM* model shares the same architecture as a single weak learner in *CoMForE*, but trained with the full dataset. The results, displayed in Table 36, present the performance of both models using (a) only trading data, (b) only sentiments, and (c) a combination of trading data and sentiments. Remarkably, the results from the sentiment data alone (b) underscore the correlation between social media sentiment and cryptocurrency prices. The results derived from the combination of trading data and Twitter sentiments (c) affirm that this combination yields the most accurate results. This suggests that social media sentiments can effectively complement trading data in capturing cryptocurrency price fluctuations, thereby enhancing the model's forecasting capability. Specifically, adding sentiment analysis to both models can improve the results by up to 15% in comparison to using only trading data.

Additionally, we performed further experiments applying *LSTM* and *CoMForE* on trading data, supplemented with various other modalities; (d) the hash rate, (e) search volume, and (f) blockchain data encompassing all six modalities. The evaluation results of both models are presented in Table 37. While the error for (d) is marginally higher for *CoMForE*, which

	(a) Only Trading		(b) Only Sentiments		(c) Trading + Sentiments	
	LSTM	*CoMForE*	*LSTM*	*CoMForE*	*LSTM*	*CoMForE*
Training RMSE ($)	346.507	**100.593***	2,617.935	**221.707**	344.082	**104.497**
Training MAE ($)	204.773	**64.651**	2,120.712	**59.85***	209.094	**61.134**
Validation RMSE ($)	**502.473**	1,097.734	**8,233.91**	8,650.681	**436.684***	1,098.085
Validation MAE ($)	**321.106**	709.154	7,210.621	**6,068.537**	**309.384***	708.485
Testing RMSE ($)	502.473	**272.027**	8,233.91	**4,006.787**	354.071	**243.47***
Testing MAE ($)	321.106	**207.332**	7,210.621	**2,969.586**	312.009	**201.568***

Table 36: Results of *LSTM* and *CoMForE* for different data modality combinations: (a) only trading data, (b) only sentiments, and (c) a combination of sentiments and trading data. The asterisk (*) highlights the lowest value for each modality, indicating the least error in each row.

is possibly due to the boosting process, the lower error observed for *LSTM* when incorporating the hash rate suggests it as a beneficial factor for a more precise price forecasting. The findings from (e) suggest that search volumes did not notably influence the performance of either *LSTM* or *CoMForE* when combined with trading data. However, it can be speculated that search volumes could exhibit enhanced performance when integrated with other modalities, such as social media sentiments, as both represent a measure of "public awareness".

The outcomes presented in column (f) for both models substantiate that incorporating blockchain data and trading data significantly enhances predictive accuracy compared to using the hash rate alone. However, the testing errors encountered with the *CoMForE* are notably larger than those found with the *LSTM* model. As demonstrated in Figure 26, when blockchain data was utilized as the sole input for both models, *CoMForE* didn't precisely forecast the price, though it could track the price trend. These results highlight a noticeable correlation between blockchain data and cryptocurrency prices, which can assist in the prediction of cryptocurrency prices. To the best of our knowledge, this is the first study to integrate blockchain data into an analytical framework for predicting cryptocurrency prices. For further insights into the interplay between different modality combinations, Figure 27 showcases the qualitative results of *CoMForE* and *LSTM* over a randomly sampled 100-day testing interval across all combinations.

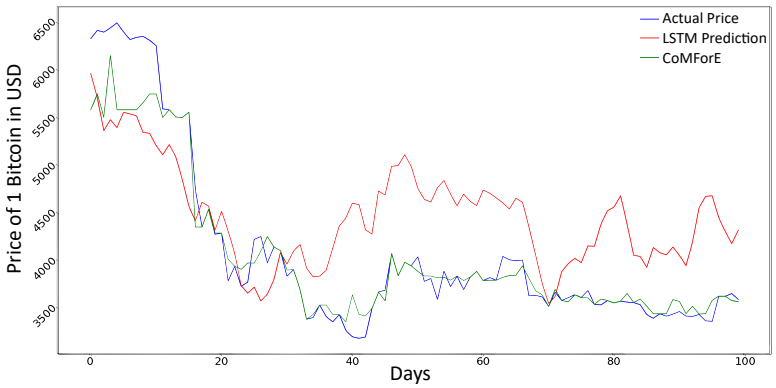


Figure 26: *LSTM* and *CoMForE* forecasting results with blockchain data over 100-day interval.

In our comprehensive evaluation, Table 38 presents the cumulative impact of combining all four modalities: trading data, social media sentiments, search volumes, and blockchain data. As clearly illustrated, this combination significantly enhances the results, indicating their complementary nature. Specifically, we observed an improvement of $75.312 in MAE, which corresponds to a 36.32% enhancement in forecasting accuracy. To visually corroborate this, Figure 28 offers a qualitative comparison over a span of 100

	(d) Trading + Hash rate		(e) Trading + Search volume		(f) Trading + Blockchain data	
	LSTM	*CoMForE*	*LSTM*	*CoMForE*	*LSTM*	*CoMForE*
Training RMSE ($)	299.818	**41.201***	352.823	**89.093**	280.944	**96.525**
Training MAE ($)	178.795	**14.433***	206.466	**30.981**	171.552	**62.029**
Validation RMSE ($)	**433.68***	1,156.835	**522.815**	1,438.16	**1,052.821**	1,151.279
Validation MAE ($)	**299.766***	782.773	**369.701**	940.658	**707.377**	765.778
Testing RMSE ($)	433.680	**356.554**	519.175	**277.861**	**200.263***	281.607
Testing MAE ($)	299.766	**291.09**	365.0	**201.177**	**156.694***	213.981

Table 37: Results of *LSTM* and *CoMForE* for trading data combined with (d) the hash rate, (e) search volume and (f) blockchain data. The asterisk (*) highlights the lowest value for each modality, indicating the least error in each row.

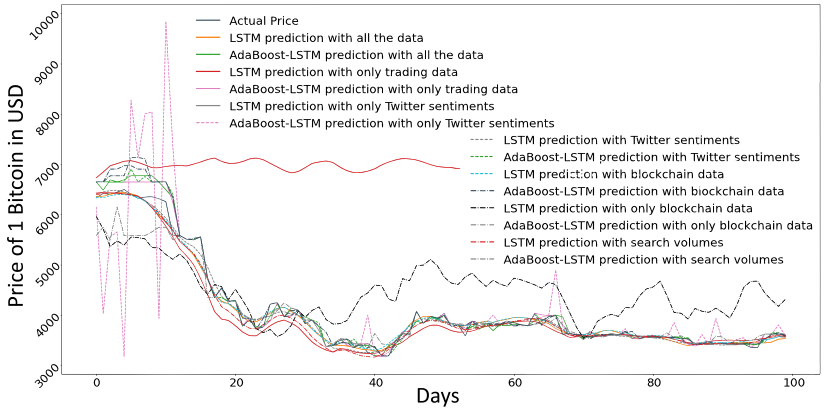


Figure 27: *CoMForE* and *LSTM* outcomes across various modality combinations over 100-day interval.

days, where the forecasted trajectories from both models closely align with the actual price trends.

	LSTM using All	*CoMForE* using All
Training RMSE ($)	280.013	**83.564**
Training MAE ($)	173.698	**25.780**
Validation RMSE ($)	389.245$	**234.718**
Validation MAE ($)	281.399	**234.666**
Testing RMSE ($)	389.245	**158.929**
Testing MAE ($)	281.399	**132.027**

Table 38: Performance of *LSTM* and *CoMForE* with all modalities incorporated.

In response to **RQ2**, Tables 36, 37, and 38 reveal the anticipated outcome, where *CoMForE* shows substantial superiority over the *LSTM* in all scenarios. In cases, (a) and (c), the *LSTM*'s overall validation error appears lower, which could be due to the inconsistent trends and volatilities in price over time. Since *LSTM* typically produces smoother forecasts than the weak learners of *CoMForE*, it yields a lower error during periods of high volatility. As depicted in Figure 24, periods from 2012 to 2017 and the end of 2018 to 2019 exhibited minor fluctuations compared to other timeframes.

Figure 29 presents the qualitative outcomes of the *CoMForE* and *LSTM* models over a randomly selected 100-day testing interval for combinations (a), (b), and (c). Similarly, Figure 30 illustrates the results for combinations (d), (e), and (f). In both figures, the blue curve represents the actual price. Meanwhile, the red and green curves depict the price forecasts generated by the *LSTM* and *CoMForE* models, respectively. Notably, *CoMForE*

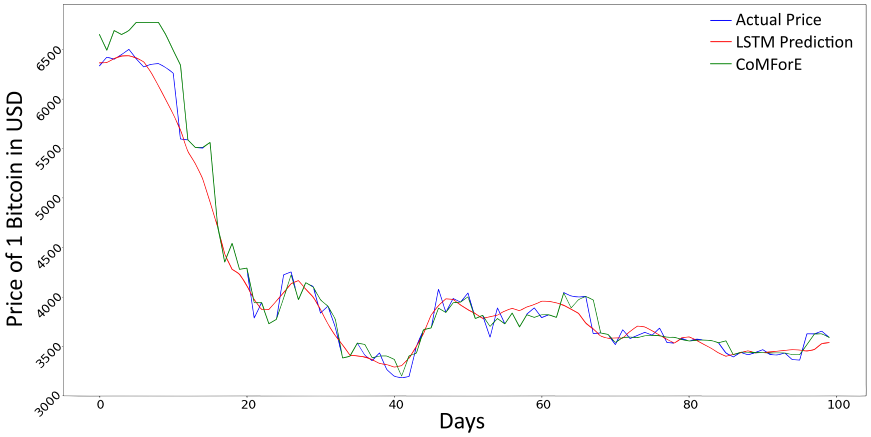
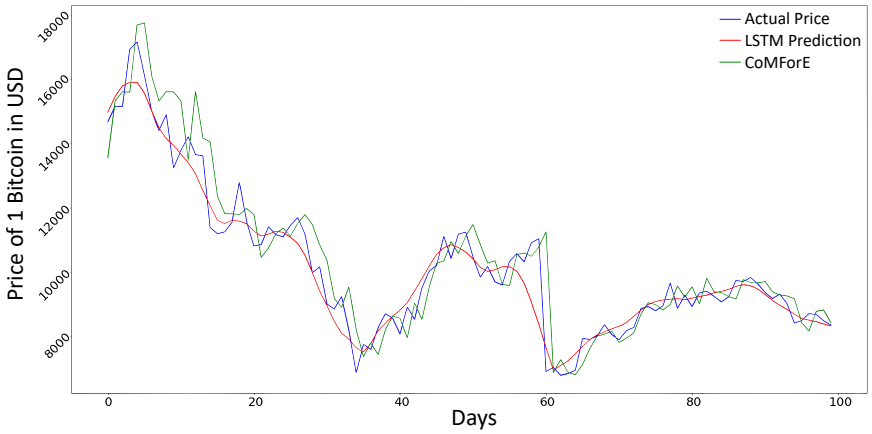


Figure 28: *LSTM* and *CoMForE* forecasting results with all modalities over 100-day interval.

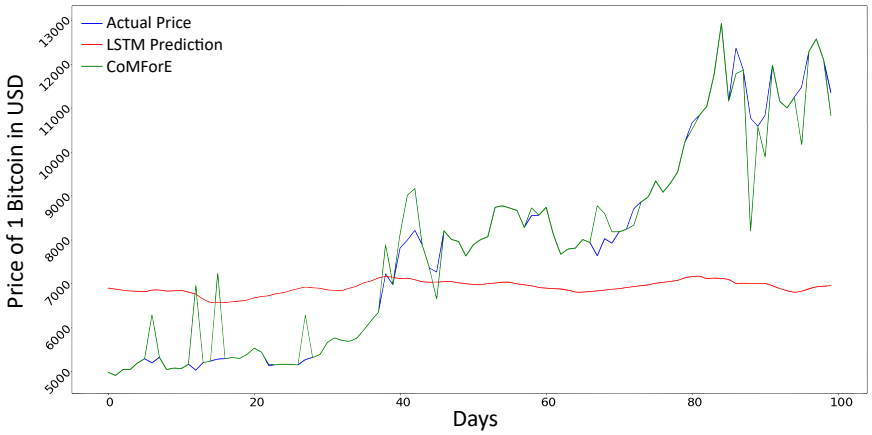
consistently outperforms the single learner, *LSTM*, especially for the combinations (a), (b), and (c). Although the forecasting error in (b) is notably higher than when only using trading data, it is significant to acknowledge that *CoMForE*, despite being fed with a solitary input per day (i.e., Twitter sentiments), managed to detect the relationship between prices and sentiments, subsequently producing a meaningful price prediction. As illustrated in Figure 29b, *CoMForE* successfully tracks the price trend using sentiment data alone. In contrast, the *LSTM* model has difficulty learning patterns from the input data and tends to produce almost constant forecasts over time.

To gain a deeper understanding of the model's performance, we computed the *MAE* distribution for the entire testing set as illustrated in Figure 31. As can be seen, the forecasted price deviates by $+/-500$\$ from the actual price of 1 ₿ approximately 68% of the time, yielding to an average error of 3.02%. This fluctuation is particularly significant given that the average ₿ price within the testing set is \$16553.59. This deviation emphasizes the commendable precision of the model's forecasting relative to the actual price.

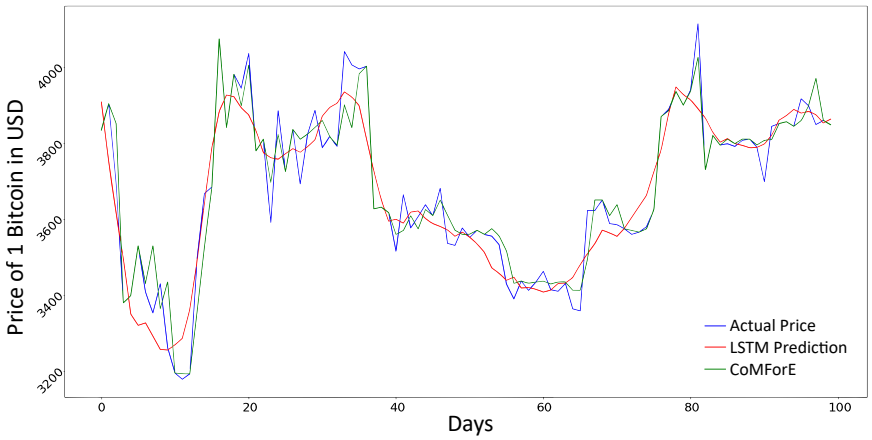
Comparison against baseline approaches Additionally, we evaluate the performance of *CoMForE*, trained with all modalities, in comparison to baseline methodologies. The comparison is conducted over the same dataset, within the period from January 1, 2020, to July 1, 2020. It is noteworthy that all models in this comparison are forecasting ₿ prices for the subsequent 24 hours. As delineated by the results in Table 39, it is evident that *CoMForE* outperforms the other models, which capitalizes on multiple data modalities and on the power of ensemble learning, enhancing the understanding of the underlying trends and thereby improving the accuracy of its predictions.



(a) Only Trading

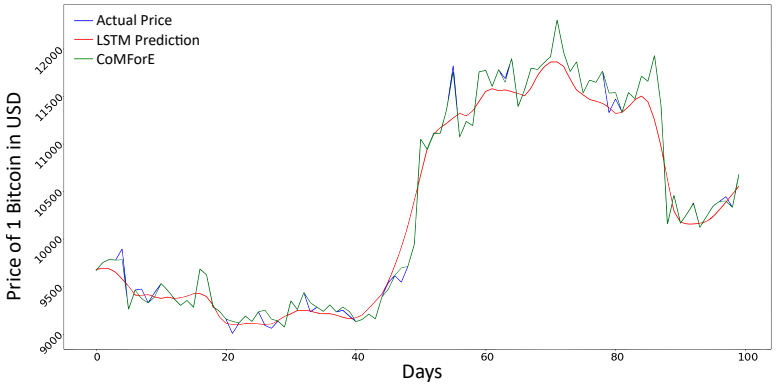


(b) Only Sentiments

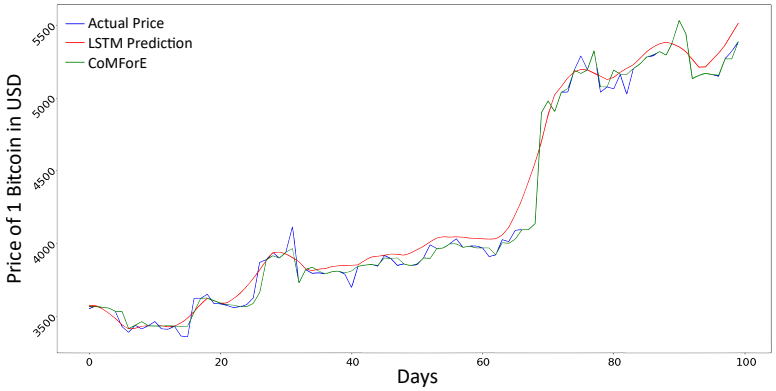


(c) Trading + Sentiments

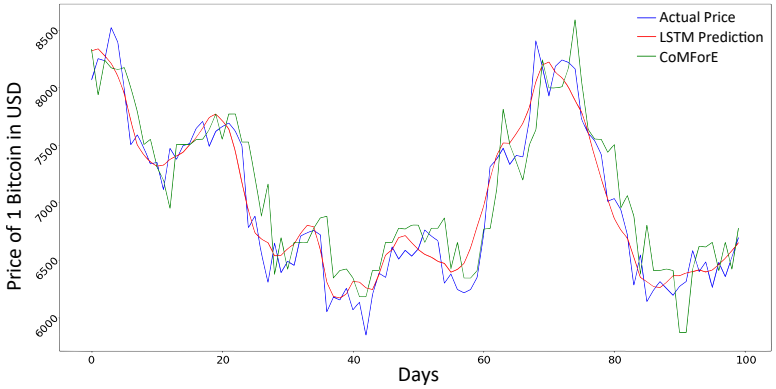
Figure 29: Visual representation of the forecasting performance of *LSTM* and *CoMForE* over a 100-day interval, tested using (a) trading data, (b) sentiment data, and (c) a combination of both.



(d) Trading + Hashrate



(e) Trading + Search volume



(f) Trading + Blockchain data

Figure 30: Visual representation of the forecasting performance of *LSTM* and *CoMForE* over a 100-day interval, tested using trading data combined with (d) the hash rate, (e) search volume, and (f) blockchain data.

Fluctuation Analysis In response to **RQ3**, we propose to equip the investor with not only the price forecast but also the range of potential price

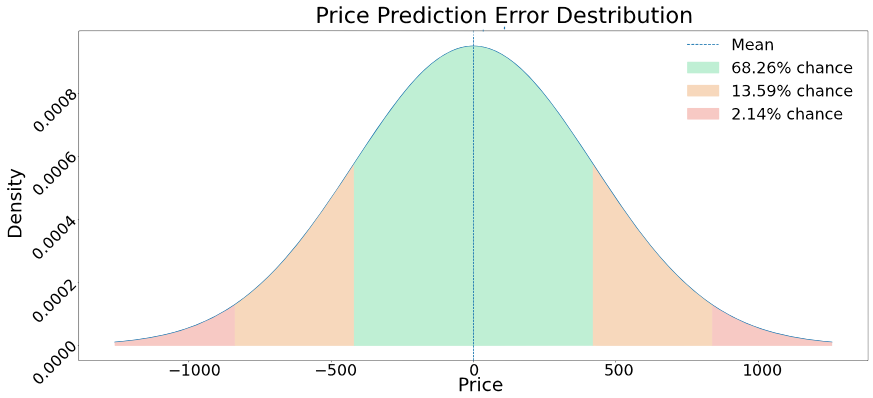


Figure 31: MAE distribution over the entire testing set for *CoMForE*.

	RMSE ($)	MAE ($)
CoMForE	**158.929**	**132.027**
BNN17	221.642	184.124
LSTM20A	166.875	148.628
LSTM20B	202.787	177.02
AdaBoost21	197.544	156.346
GRU21	208.006	160.44
ARIMA19	584.384	542.73
GRU20	185.154	157.632

Table 39: Evaluation Comparison Between the results of *CoMForE* and baseline approaches on test data.

fluctuations. Given the complexity of quantifying this, we present in Figure 32 a set of outcomes obtained using the approach described in Section 8.3.2 over four random days. As illustrated, the certainty of price forecast can vary day-to-day. On days marked by significant volatility, the price distribution appears flat and exhibits a high standard deviation, indicating a high likelihood of deviation from the model's singular prediction. Conversely, on days when the model predicts with a high degree of certainty, the price distribution tends to peak sharply and display a lower standard deviation. These prediction distributions serve as a valuable resource for investors as they offer clear insights into the model's confidence in its predictions for a specific day.

Long-term forecasting In light of the promising obtained results, we further explored the limitations and performance of *CoMForE* and multi-modality combination in generating long-term predictions. Therefore, we continuously predicted the price by appending the last prediction to the input of the prediction of the subsequent day. This process is reiterated for

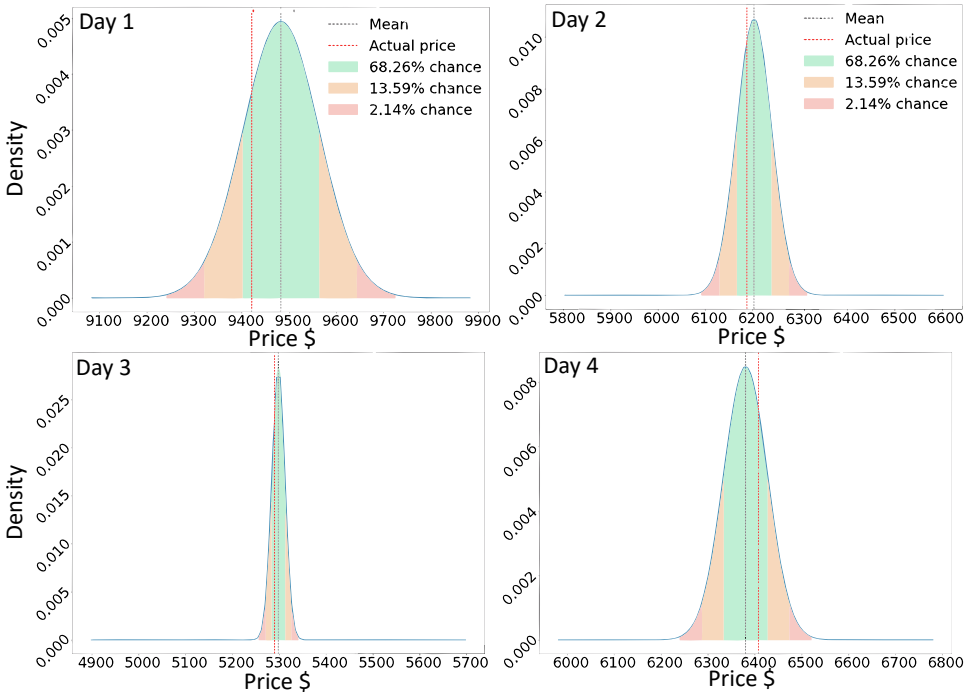


Figure 32: Predicted price distribution of four randomly selected days. *Mean* denotes the predicted price

up to the 30th day. It is crucial to note that only historical trading data is considered since other data modalities cannot be predicted. This assumption implies a likely performance dip because the model is model operating with less data for each successive day's prediction. Figure 33 showcases the results of this experiment, with the red curve representing the mean MAE of each time-window prediction. The boundaries of the blue area, meanwhile, indicate the highest and lowest MAE. As anticipated, the predictive accuracy is at its peak on the first day and progressively diminishes for longer-term forecasts.

8.5 Conclusion

In this paper, we explored the impact of various independent factors on cryptocurrency volatility through a comprehensive multimodal forecasting approach, aiming to provide investors with accurate short-term price forecasts based on correlated sources (**RQ1**). The proposed approach harnesses the power of ensemble learning, combining multiple weak LSTM learners (**RQ2**. Understanding that investors need a more comprehensive view of the market, we introduced the concept of fluctuation distributions, which offer a broader perspective on the market and provide insights into the reliability of the price forecast (**RQ3**).

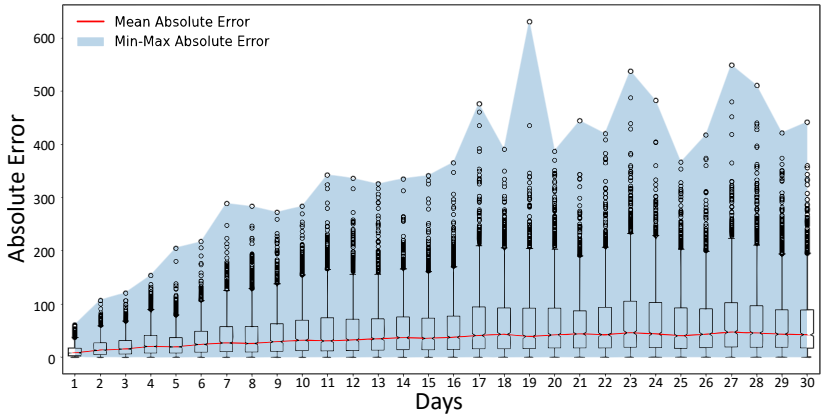


Figure 33: Bitcoin price prediction absolute testing errors over different time windows

For future work, we plan to expand the application of this approach to other cryptocurrencies, incorporating the Bitcoin price as a vital variable. We also propose to generate and train the model on synthetic trading data characterized by extremely high volatility. This approach stems from our hypothesis that preparing the model for unprecedented market volatility will enhance its adaptability to future surprises inherent in the cryptocurrency markets.

9 Paper 10: Visual Question Answering

COIN: Counterfactual Image Generation for Visual Question Answering Interpretation

Zeyd Boukhers(✉), *Timo Hartmann, Jan Jürjens*

(DOI: 10.3390/s22062245)

Abstract Due to the significant advancement of Natural Language Processing and Computer Vision-based models, Visual Question Answering (VQA) systems are becoming more intelligent and advanced. However, they are still error-prone when dealing with relatively complex questions. Therefore, it is important to understand the behaviour of the VQA models before adopting their results. This paper introduces an interpretability approach for VQA models by generating counterfactual images. Specifically, the generated image is supposed to have the minimal possible change to the original image and lead the VQA model to give a different answer. In addition, our approach ensures that the generated image is realistic. Since quantitative metrics cannot be employed to evaluate the interpretability of the model, we carried out a user study to assess different aspects of our approach. In addition to interpreting the result of VQA models on single images, the obtained results and the discussion provide an extensive explanation of VQA models' behaviour.

Keywords: *ML interpretability; VQA; GAN; UXE*

9.1 Introduction

Over the past years, the task of Visual Question Answering (VQA) has been widely investigated taking advantage of the development strides of Natural Language Processing (NLP) and Computer Vision (CV). A VQA model aims to answer a natural language question about the content of an image or one of the appearing objects. Due to the complexity of the task, VQA systems are still in the early stage of research and up to our knowledge, they are not integrated into any running system. One of the inherent problems of VQA systems is the reliance on the correlation between the question and answer more than the content of the image [73, 260]. Furthermore, the available datasets are usually unbalanced w.r.t certain types of questions. In the VQAv1 dataset [19], for instance, simply answering *tennis* to any sports-related question without considering the image yields an accuracy of approximately 40%. This is because the dataset creators tend to generate questions about objects detectable in the image which make the dataset suffer from the so-called "*visual priming bias*". For example, blindly answering "*yes*" to all questions starting with "*Do you see a...?*" without considering anything else yields approximately 90% accuracy in the VQAv1 dataset [260,

128]. In practice, this bias is not distinctly perceivable because the users tend to ask similar questions related to the image or the appearing objects and they most likely know the correct answer. For more complex questions or when the user lacks the knowledge expertise of the questions or the image content (e.g. medical domain), it won't be possible to capture the behaviour of the VQA system and whether it is biased. Therefore, it is important to interpret the result of these systems and find what caused the model to output an answer based on the image-question pair.

Although there is no uniform definition of "*Interpretability*", researchers agree that the interpretability of ML models increases the users' trust in ML systems. For VQA models, there exist a limited number of papers that investigate the interpretation of their models. Existing attempts include (i) the identification of visual attributes that are relevant to the question [391, 215] and (ii) the generation of counterfactuals [260, 268, 339].

In the direction of (i), the method proposed by Zhang et al. [391] generates a heatmap over the input image to highlight the image regions that are relevant to the answer of the VQA system. However, as pointed out by Fernández-Loría et al. [111], this approach explains the system's prediction but does not provide a sufficient explanation of its decision. They suggest instead that counterfactual explanations offer a more sophisticated way to increase the interpretability of an ML model because they reveal the causal relationship between features in the input and the model's decision. For example, considering an ML model that classifies MRI images into *Malignant* or *Benign*, the generated heatmap can highlight the key Region Of Interest (ROI) of the model's prediction. Although this heatmap answers the question *What did lead to this decision?*, it does not answer another important question: *Why did it lead to this decision?* Consequently, this approach does not provide insight into how the model would behave under alternative conditions. To provide more in-depth interpretability, generating a counterfactual image that is minimally different from the original one but leads to a different model's output would indirectly answer the question *Why did the model take such a decision?* To the best of our knowledge, only three existing methods [73, 268, 339] aim at making VQA models interpretable by providing counterfactual images.

Chen et al. [73] introduce a method that generates counterfactual samples by applying masks to critical objects in the images or questions' words. Similarly, Teney et al. [339] present a method that masks features in the images whose bounding boxes overlap with human-annotated attention maps. Finally, in their ongoing research, Pan et al. [268] propose a framework to generate counterfactual images by editing the original image such that the VQA system returns an alternative answer for a given question. Due to the complexity of the problem, the approach is restricted to colour-based questions. Given a tuple (Image, Question and the VQA's answer), the approach first finds the question-critical object and then changes its colour so that the

VQA system gives a different answer. However, this change is not limited to the question-critical object but all regions with similar colours are changed. Given that the main goal of interpreting VQA systems is to help the user understand the behaviour of the VQA model, the approach presented in [268] requires that the user understands the relationship between the image, the question and the answer.

As the user needs interpretation mainly when he lacks necessary knowledge to understand the relationship between the input and output, this paper aims to best interpret the output of VQA systems by generating counterfactual images that lead the VQA model to either (1) output a different answer or (2) deviate its focus on another region. Specifically, this paper aims to answer the following research questions:

- **RQ1:** How to change the answer of a VQA model with the minimum possible edit on the input image?
- **RQ2:** How to alter exclusively the region in the image on which the VQA model focuses to derive an answer to a certain question?
- **RQ3:** How to generate realistic counterfactual images?

To this end, we propose to extend the work proposed Pan et al. [268] by restricting the changes to the question-critical region. Specifically, this paper introduces an attention mechanism that identifies the question-critical region in the image and guides the counterfactual generator to apply the changes on those regions. Moreover, a weighted reconstruction loss is introduced in order to allow the counterfactual generator to make more significant changes to the question-critical ROI than the rest of the image. For further improvement and future work, we made the entire implementation of the guided generator publicly available.

Following this section, Section 9.2 discusses the related works. Section 9.3 presents the proposed approach and Section 9.4 presents the conducted experiments and the obtained results that validate the effectiveness of the proposed approach. Finally, Section 9.5 concludes this paper and gives insight into future directions.

9.2 Related Work

This paper addresses the problem of interpreting the outcome of VQA systems. Therefore, we will review in this section the related works divided into three categories: (1) Interpretable Machine Learning, (2) Visual Question Answering (VQA) and (3) Interpretable VQA.

9.2.1 Interpretable Machine Learning

Throughout the past decades, the notion of interpretability increasingly gained attention by the Machine Learning (ML) community [64, 99, 125, 186, 317, 346, 388, 389]. According to Kim et al. [186], interpretability is particularly important for systems whose decisions have a significant impact such as in healthcare, criminal justice and finance. Interpretability serves several purposes, including protecting certain groups from being discriminated against, understanding the effect of parameter and input variation on the model's robustness and increasing the user's trust in automated intelligent systems [99]. Therefore, a model is considered interpretable if it allows a human to consistently and correctly classify its outputs [186] and understand the reason behind the model's output [64]. ML models such as decision trees are inherently interpretable, as they provide explanations during training or while the output is generated. However, most of the sophisticated ML models used nowadays such as Deep Neural Networks are not interpretable by nature and require post-hoc explanations [245]

One way to create such explanations is to use global surrogate models, where the aim is to approximate the prediction function f of a black-box and complex model (e.g. neural networks) to the best possible with a prediction function g such that g is the prediction function of an inherently interpretable model (e.g. decision trees or linear regression) [261, 245]. Another way is to use local surrogate models, which individually explain the predictions of a trained ML model to have an overview of its behaviour [261]. Ribeiro et al. [299] propose a Local Interpretable Model-agnostic Explanations (LIME) which approximates the output of black-box models by examining how variations in the training data affect its predictions. Particularly, LIME permutes a trained black-box model's training samples to generate a new dataset. Based on the black-box model's predictions on the permuted dataset, LIME trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest.

Feature visualization is another direction to increase the interpretability of black-box models. The goal is to visualize the features that maximize the activation of a NN's unit [261]. This direction takes advantage of the structure of NNs, where the relevance is backpropagated from the output layer to the input layer [388]. Mainly, most of the approaches under this direction are dedicated to image classification tasks by providing a saliency map that highlights the pixels relevant to the model's output [243, 245, 319]. A common interpretability direction suggests generating example-based explanations for complex data distributions. The aim is to find prototypes from the training dataset that summarize the prediction of the model [245]. Although this approach can satisfy the user need for interpretation in simple tasks, it is not practical for most of the real-world data which are heavily complex and seldom contain representative prototypes [186]. Therefore, Kim

et al. [186] propose to identify some criticism samples that deliver insights about those prototypes which is not covered by the model.

The problem of post-hoc interpretability methods is their incapability to answer how the model would behave under alternative conditions (e.g. different training data). Therefore, causal interpretability approaches aim at finding why did the model make a certain decision instead of another one or what would be the output of the model for a slightly different input [245]. Here, the goal is to extract causal relationships from the data by analysing whether changing one variable cause an effect in another one [249]. One way to achieve this is by finding a counterfactual input that affects the model's prediction [92, 126, 129, 150, 322, 351]. For instance, Goyal et al. [129] propose a method that identifies how a given image "I" could change so that the image classifier outputs a different class by replacing the key discriminative regions in "I' with pixels from an identified "distractor" image "I_0' that has a different class label. Pearl [272] suggests that generating counterfactuals allows for the highest degree of interpretability among all methods to explain black-box models.

9.2.2 Visual Question Answering (VQA)

Taking advantage of the remarkable advancement of Computer Vision, Natural Language Processing and Deep Neural Networks, several research works have addressed the task of VQA in the past years [73, 128, 19, 233, 325, 387, 18, 123, 138]. According to Antol et al. [19], VQA methods aim at answering natural language questions about an input image. This combination of image and textual data makes VQA a challenging multi-modal task that involves understanding both the question and the image [325]. The answer's format can be of several types: a word, a phrase, a binary answer, a multiple-choice answer, or a "fill in the blank" answer [325, 138].

In contrast to earlier contributions, recent VQA approaches aim at generating answers to free-form open-ended questions [364]. Agrawal et al. [19], for example, propose a system that classifies an answer to a given question about an image by combining a Convolutional Neural Networks (CNN)-based architecture to extract features from the image and Long Short-Term Memory (LSTM)-based architecture to process the question. This model, referred to as *Vanilla VQA*, can be considered as a benchmark for DL-based VQA methods [325]. Yang et al. [378] introduce *Stacked Attention Networks (SANs)* that uses CNNs and LSTMs to compute an images' regions related to the answer based on the semantic representation of a natural language question. Similarly, Anderson et al. [18] propose to narrow down the features in images by using top-down signals based on a natural language question to determine what to look for. These signals are combined with bottom-up signals stemming from a purely visual feed-forward attention mechanism.

Despite the continuous advancements in VQA, several papers suggest

that VQA systems tend to suffer from the language prior problem, where they tend to achieve good superficial performances but do not truly understand the visual context [73, 128, 387, 398]. Specifically, Goyal et al. [128] found that in the VQAv1 dataset [19] blindly answering "*yes*" to any question starting with "*Do you see a...?*" without taking into account the rest of the question or the image yields an accuracy of 87%. To overcome this, they proposed a balanced dataset to counter language biases, such that for a given triplet (image I; question Q; answer A) from the VQAv1 dataset [19], humans were asked to identify a similar image I' for which the answer to question Q is different from A. Similarly, Zhang et al. [387] propose a balanced VQA dataset for binary questions, where for each question, pairs of images showing abstract scenes were collected so that the answer to the question is "*yes*" for one image and "*no*" for the other.

Moreover, Chen et al. [73] assume that existing VQA systems capture superficial linguistic correlations between questions and answers in the training set and, hence, yields low generalizability. Therefore, they propose a model-agnostic Counterfactual Samples Synthesizing (CSS) training scheme that aims at improving VQA systems' visual-explainable and question-sensitive abilities. The CSS algorithm masks (i) objects relevant to answering a question in the original image to generate a counterfactual image and (ii) critical words to synthesize a counterfactual question. In the same context, Zhu et al. [398] propose a self-supervised learning framework that balances the training data but first, identifies whether a given question-image pair is relevant (i.e., the image contains critical information for answering the question) or irrelevant. This information is then fed to the VQA model to overcome language priors.

These above-discussed problems indicate that the empirical results of VQA systems do not reflect their efficacy, especially when promoting VQA systems to serve their intended purposes. Specifically, answering questions that the user cannot answer, such as in the healthcare domain. Therefore, it is important to make the output of the VQA model interpretable and not only rely on the evaluation results.

9.2.3 Interpretable VQA

In most real-world scenarios, human users want to get an explanation along with a VQA system's output, especially if it fails to answer a question correctly or when the user does not know whether the answer is correct [215]. However, there exist only a few papers addressing the task of interpreting and explaining the outcome of VQA systems [260, 86, 215, 268, 391]. Also, most of the existing approaches rarely provide human-understandable explanations regarding the mechanism that led to a given answer. Li et al. [215] introduce a method that simulates the human question-answering behaviour. First, they apply pre-trained attribute detectors and image captioning to ex-

tract attributes and generate descriptions for the given image. Second, the generated explanations are used instead of the image data to infer an answer to a question. Consequently, providing critical attributes and captions to the end-user allows them to understand better what the system extracts from the image. Zhang et al. [391] introduce a heat map-based system to display the image's regions relevant to the question to the user. To this end, they employed in their model region descriptions and object annotations provided in the Visual Genome dataset [196].

Pan et al. [268] introduce a method that provides counterfactual images along with a VQA model's output. Precisely, for a given question-image pair, the system generates a counterfactual image that is minimally different from the original image and visually realistic but leads the VQA model to output a different answer for the given question. In its current form, their method is restricted to the context of colour questions. Furthermore, since their model makes edits on a pixel-by-pixel level, the counterfactual images contain changes also in areas irrelevant to a given question. Therefore, the counterfactual image does not provide any meaningful interpretability if the question-critical object has the same relative colour as other objects.

9.3 Method

In this paper, we propose a method named *COIN* that provides human-interpretable discriminatory explanations for VQA systems. The aim is to interpret the outcome of the VQA system by answering the question: "*How would the image look like so that the VQA system gives a different outcome?*". Concretely, given an image-question pair $(I; Q)$ and a VQA model $f : (I, Q) \rightarrow A$, where A is its predicted answer, the goal is to train a model $\mathcal{G}$ to generate a new image I', such that:

$$\mathcal{G} : (I, Q, A) \rightarrow I' \quad | \quad f : (I', Q) \rightarrow A' \quad ; \quad \forall A' \neq A, \tag{48}$$

where I' is the counterfactual image of I. Since there are infinite possible images that can satisfy the above constraint, along the lines of Pan et al. [268], *COIN* aims to tackle the research question ***RQ1*** by constraining $\mathcal{G}$ to (i) be different as minimally as possible from I, (ii) be visually realistic, (iii) contain semantically meaningful edits and (iv) be applied only on the question-relevant regions. Intuitively, with these constraints, we aim to change the output of the VQA system by applying as minimum as possible changes only on the semantically relevant object so that the user can perceive what can change the output of the VQA system. To this end, we propose to extend the counterfactual GAN introduced by Pan et al. [268] by tackling, in addition to color-based questions, shape-based questions and ensuring that only the question-critical regions in an image are altered while retaining the rest of the image. The variables used for the system definition are summarized in Table 40.

Variable	Description
$\mathcal{G}$	The counterfactual generator proposed in this paper.
f	The VQA system (i.e. MUTAN [33] in this paper)
I	Original image
Q	Question about I
A	The answer of f to Q given I
h	I's height
w	I's width
I'	The counterfactual image of I, generated by $\mathcal{G}$
A'	The answer of f to Q given I'
$\hat{I}$	An image generated by $\mathcal{G}$
M	The attention map of I
M'	The attention map of I'

Table 40: Summary of variables used in this paper

Figure 34 illustrates an overview of the proposed architecture. In the depicted example, given the image I, the answer of the VQA system to the question " *Q: What color is the large central flower?*" is " *A: yellow*". To explain this output, $\mathcal{G}$ goes through several components:

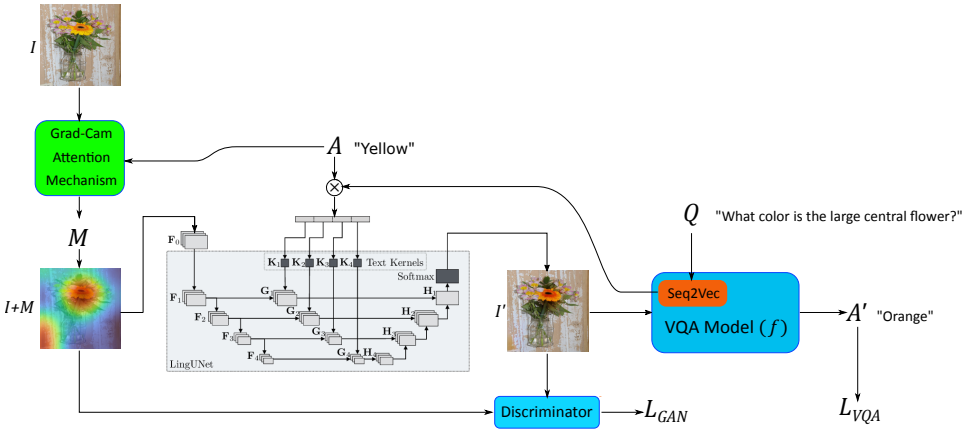


Figure 34: Overview of the proposed architecture inspired by Pan et al. [268]

9.3.1 ROI Guide

To tackle the research question ***RQ2***, $\mathcal{G}$ has to be guided to primarily edit regions in I that are relevant to Q. To this end, *COIN* aims to identify the question-critical ROI in I, but, complex images may contain various objects, of which, usually, only one or a few are relevant when answering a given question. Therefore, an object or a region can be considered to be question-

critical if it is key to finding an answer to a given question. For example, given the question "*What colour are the man's shorts?*", the question-critical object in Figure 35 is the man's shorts. Therefore, *COIN* aims to guide the generator with a continuous attention map $M \in \mathbb{R}^{1\times h\times w}$ in the range $[0, 1]$, where h and w correspond to the height and width of the input image, respectively. This map is supposed to highlight the discriminative ROIs of the image I that led $f(I)$ to output the answer A. Thus, instead of generating a counterfactual image I' based on the original image I, the latter is concatenated with the attention map M, such that the concatenation $[I; M]$ serves as an input to the generator $\mathcal{G}$.

To obtain M, an attention mechanism is used to determine each pixel's importance w.r.t the VQA system's decision. The intuition is to identify the spatial regions in an image that are most relevant to answer a given question. For this reason, *COIN* applies the Gradient-weighted Class Activation Mapping (Grad-CAM) algorithm [311] to the VQA system's final convolution layer because convolution layers retain spatial information that is not kept by fully connected layers. Specifically, CNN's last convolution layer is supposed to have the finest balance between high-level semantics and fine-grained spatial information. Grad-CAM exploits this property by finding the gradient of the most dominant logit (i.e., in the case of a VQA system, this corresponds to the answer with the highest probability) that flows into the model's final activation map. Furthermore, since Grad-CAM is suitable for various CNN-based models, it can be applied to most VQA systems.

Intuitively, the algorithm computes the importance of each neuron activated in the CNN's final convolutional layer with respect to its prediction. Computing the gradient y^a of the logit corresponding to the VQA system's predicted answer a with respect to the kth feature map's activations ϕ^k of a convolutional layer, i.e. $\frac{\delta y^a}{\delta \phi^k}$, reveals the localization map $L^c_{Grad-CAM} \in \mathbb{R}^{u\times v}$ of width u and height v. Next, channel-wise pooling with respect to the width and height dimensions is applied to the gradients. The pooled gradients are then used to weigh the activation channels. Finally, the weighted activations α^a_k reveal each channel's importance with respect to the VQA system's prediction [311]:

$$\alpha^a_k = \frac{1}{Z}\sum_i^u \sum_j^v \frac{\delta y^a}{\delta \phi^k_{i,j}} \tag{49}$$

Performing a weighted combination of forward activation maps followed by a ReLU finally yields a coarse saliency map of the same size as the convolutional feature maps [311]:

$$L^c_{Grad-CAM} = ReLU\left(\sum_k \alpha^a_k \phi^k\right) \tag{50}$$

Figure 35: Example question-image pair from the VQAv1 dataset [19]. The red bounding box indicates the question-critical object.

Finally, to obtain M, the feature maps $L^c_{Grad-CAM}$ are interpolated to match the size of the input image I. Furthermore, a gaussian filter with a mean $\mu = 0$ and a population standard deviation $\sigma = 2$ is applied for improved preservation of the selected image regions' edges [34].

9.3.2 Language-Conditioned Counterfactual Image Generation

To drive $\mathcal{G}$ to generate a counterfactual image I' such that the corresponding answer $A' \neq A$, *COIN* follows Pan et al. [268] by adopting an architecture based on LingUNet [239], which is an encoder-decoder Neural Network (NN) similar to the popular pixel-to-pixel UNet model [303]. LingUNet maps conditioning language to key intermediate filter weights based on an embedding of natural language text.

Similarly to [268], *COIN* feed $\mathcal{G}$ with language embedding which is the concatenation of the VQA system's question encoding and answer encoding. The question encoding is represented by the question embedding $\overline{q}$, which stems from the VQA system's language encoding for the question Q. The answer encoding is represented by the answer embedding $\overline{a}$, which is the VQA's final logits weight vector w.r.t its prediction A for the image-question pair (I, Q). The goal here is to train $\mathcal{G}$ with the VQA system's negated cross-entropy for A being the target. Consequently, the generated image I' should contain semantically meaningful differences compared to I, such that the VQA system outputs two different answers for I' and I given the same question Q.

Precisely, $\mathcal{G}$ applies a series of operations to condition the image generation process on language. First, the question embedding $\overline{q}$ and the answer embedding $\overline{a}$ are concatenated to create a language representation $\overline{x}$. Sec-

ond, $\mathcal{G}$ applies a 2D 1×1 convolutional filter with weights K_k to each feature map F_j . Each K_k is computed by splitting $\overline{x}$ into m equally sized vectors $\{\overline{x}\}_{j=1}^{m}$ and applying a 1×1 linear transformation to each of them. Applying the filter weights to each F_j yields the language-conditioned feature maps G_j [239].

Next, LingUNet performs a series of convolution and deconvolution operations to generate a new image $\hat{I}$. The final counterfactual I' is retrieved as follows:

$$I' = M \odot \hat{I} + (\mathbf{1} - M) \odot I, \tag{51}$$

where $\odot$ denotes the element-wise multiplication and $\mathbf{1}$ is an all-ones matrix with the same dimension. Intuitively, I' is created by incorporating to the original image's background, the foreground of $\hat{I}$, which is denoted by pixels with large attention values representing a higher intensity compared to those with low attention values.

9.3.3 Minimum change

Although I and I' should have distinct semantics with respect to a given question Q, the differences between the two images should be as minimal as possible. To this end, *COIN* incorporates a reconstruction loss, which penalizes the generator for creating outputs different from the input. To ensure that question-critical objects can change their semantic meaning, the generator should be allowed to make significantly more changes in the corresponding image regions (i.e. the foreground) than in the rest of the image (i.e., the background). A modified ℓ_2-loss adapted to this purpose, which incorporates the attention map M as a relative weighting term, acts as the reconstruction loss:

$$\ell_2 = \left[|| \left(\mathbf{1} - M \right) \odot I - \left(\mathbf{1} - M \right) ||_2^2 \right]. \tag{52}$$

Applying a weighted reconstruction loss aims at contributing to the desired traits that (i) the model predominantly edits critical objects and (ii) a relatively ℓ_2-loss constraint is applied to question-critical regions, allowing for more significant semantic edits. Contrarily, the stricter ℓ_2- constraint for question-irrelevant regions ensures that the generator retains them nearly unchanged.

9.3.4 Realism

The counterfactual images generated by $\mathcal{G}$ should be visually realistic. To this end and to tackle the research question (***RQ3***), *COIN* employs a PatchGAN discriminator as proposed by Isola et al. [170]. This discriminator learns to distinguish between real and fake images and penalizes unrealistic

generated counterfactual images. The generator and the discriminator are trained in an adversarial manner as in GAN training [127].

Spectral Normalization for stabilize training: Training GANs can suffer from instability and be vulnerable to the problems of exploding and vanishing gradients [218]. In their approach, Pan et al. [268] applied gradient clipping to counter this problem, which requires extensive empirical fine-tuning of the training regime. To bypass this extensive procedure, *COIN* uses spectral normalization [218, 240] to counteract training instability as Miyato et al. [240] suggest that using spectral normalization in GANs can lead to the generated images having a higher quality relative to other training stabilization techniques, such as gradient clipping. Given a real function $\gamma : \mathbb{R} \rightarrow \mathbb{R}$, the Lipschitz constraint is followed if $|\gamma(x_i) - \gamma(x_2)|/|x1 - x2| \leqslant k$, where k is the Lipschitz constanz (e.g., $k = 1$). Given a CNN $\mathcal{CNN}_\theta$ with L layers and weights $\theta = \{w_1, w_2, \cdots, w_L\}$, its output for an input x can be computed as [218]:

$$\mathcal{CNN}_\theta = a_L \odot l_{w_L} \odot a_{L-1} \odot l_{w_{L-1}} \odot \cdots \odot a_1 \odot l_{w_1}(x), \tag{53}$$

where $a_{i=1}^{L}$ denotes the activation function in the ith layer. Spectral normalization regularizes the convolutional kernels $w_i \in \mathbb{R}^{c_{out} \times c_{in} \times k_w \times k_h}$ with kernel width k_w and height k_h of the fully connected layers l_{w_i} and c_{in} and c_{out} be the input and output channels, respectively. To this end, w_i is first reshaped into a matrix $\hat{w}_i \in \mathbb{R}^{c_{out}} \times (c_{in} \times k_w \times k_h)$, which is then normalized such that the spectral norm $||\hat{w}_i||_{sp} = 1 \forall i = 1, \cdots, L$. Thereby, the spectral norm is computed as follows [218]:

$$||\hat{w}_i||_{sp} = \frac{\hat{w}_i}{u_i^T \times \hat{w}_i \times v_i}, \tag{54}$$

where u_i and v_i denote the left and sight singular vectors of $\hat{w}_i$ with respect to its largest singular value.

9.4 Experiments

In this section, we evaluate the effectiveness of the proposed approach from different aspects, namely, the capability of $\mathcal{G}$ to (i) generate counterfactual image I' such that $f(I'; Q) \neq f(I; Q)$ (***RQ1***), (ii) focus the changes on the question-critical region (***RQ2***), (iii) generate realistic images (***RQ3***). For result reproducibility and further improvements, we made our code and results publicly available under this link [48]

[48] `https://coin.ai-research.net/`

9.4.1 Dataset

For all our experiments, we used a subset of the VQAv1 dataset's *Real Images* portion introduced by Agrawal et al. [19] The dataset covers images of everyday scenes with a wide variety of questions about the images and the corresponding ground truth answers. Despite the dataset including samples with several types of questions, for feasibility reasons, we focus in this experiment on color- and shape-based questions only. This yields a set of 23,469 tuples (Image, Question, Answer). The subset is publicly available for further improvements.[48]

9.4.2 VQA system

In our experiments, we employed MUTAN [33], which is trained on VQAv1 dataset. MUTAN achieved an overall accuracy of approximately 67% and it performed particularly well on questions with binary *Yes/No* answers (accuracy $\approx$ 85.14%). For quantitative questions (e.g., "*How many ...?*"), it achieved an accuracy of 39.81%. For all other question types, including color and shape-based questions, it achieved an accuracy of 58.52%.

9.4.3 Evaluation and Results

Automatically and objectively assessing the quality of synthetically generated images is a challenging task [252, 304, 397]. Salimans et al. [304] suggest that there exists no objective function to assess a GAN's performance. Furthermore, the goal of *COIN* is to provide an understandable interpretability to the VQA output. The quality of this interpretability can only be assessed by the satisfaction of the user. Therefore, we conducted a user study by presenting the output of *COIN* (i.e. I' and A') together with I, A and Q to the participants. For every sample, the user answers five questions divided in two phases:

- **Phase I:** We present the participant with I', Q and A'. The participant is requested then to answer the following questions:
  1. *Is the answer correct?*
     with three possible answers: *Yes*, *No*, and *I am not sure.*
  2. *Does the picture look photoshopped: any noticeable edit or distortion (automatic or manual)?* with five possible answers (i.e. from Very real to Clearly photoshopped).
- **Phase II:** We present the participant with Q and both images I and I' together with the answers of the VQA system A and A'. The participant is requested then to answer the following questions:

1. *Which of the images is the original?* with three possible answers: *Image 1*[49], *Image 2*[49] and *I am not sure*
2. *Is the difference between both images related to the question-critical object?* with three possible answers: *Yes*, *No* and *I am not sure*
3. *Which pair (Image, Answer) is correct?* with four possible answers: *Image 1*[49], *Image 2*[49], *Both* and *None*

To make this experience easy for the participants, we built a web application [48], where 94 participants have participated in the survey answering the above questions for 1320 unique samples. Note that some samples have been treated by more than one participant (maximum three), which make the total number of samples 2001. In the following, we present the qualitative and quantitative (obtained from the user study) results of *COIN* w.r.t the above-mentioned evaluation aspects:

Semantic change (RQ1): The main goal of *COIN* is to interpret the result of VQA systems by trying to generate images with the minimum possible change from the original ones so that the VQA system changes its answer. Therefore, we evaluate here the capability of $\mathcal{G}$ to generate these images. Among 12096 counterfactual images generated by $\mathcal{G}$, 37.82% of them lead f to output an answer $A' \neq A$. In particular, f outputs a different answer for 38.05% of the color-based questions and for 25.45% of the shape-based questions. This can be caused by several reasons, such as:

- The question-critical region is very large but the VQA system focuses on a very small region. Once altering that region, the VQA system slightly deviates its focus to another region (see the example in Figure 38b and result discussion in **RQ2**). Although the answer does not change, it interprets the outcome of the VQA system and its behaviour. Specifically, why the model outputs the answer A and whether the model sticks to a specific region for answering a question Q.
- The image requires a significant change so that the answer is changed but due to the other constraints (e.g. minimum change, realism, etc), the generator cannot alter the image more. Here also, the interpretation would be that the VQA system is confident about the answer and a lot of change is required to change its answer.
- The VQA system does not rely on the image while deriving the answer (see Section 9.2.2).

Among the samples treated in our survey, the participants found that the VQA outputs a correct answer A given I and Q for only 45.8% of the

[49]Note that we do not which of the images is I and which one is I'

presented images, while it outputs a wrong answer for 41.4% of them. In the remaining 12.8% of the images, the participants couldn't decide because (i) the question was not understood, (ii) the correct answer is not unique or (iii) the answer is only partially correct. After presenting the participants with both images I and I', the question Q and the answers A and A', the participants changed their opinions about 484 samples, where they found that A is correct for 70.5% of the presented images. This indicates that the participants could understand the question and answer better after interpreting the result of f. For the generated images, the participants found that f outputs a correct answer A' for only 40.5%.

Figure 36 illustrates qualitative results of $\mathcal{G}$ for color-based questions, where each row represents, from left to right, the original image I, the corresponding map M and the generated counterfactual image I' and its corresponding map M'. The rows 36a to 36c show examples of counterfactual images with different answers than their originals with realistic and understandable changes. As can be noticed in some examples such as Figures 36a and 36b, the VQA system f slightly shifts its attention after the change is applied. This means, that theoretically, f can give a different answer because of focusing on another region after the edit and not because of the edit itself.

For the rest of the samples, f fails to alter their semantic meaning. Figure 36d depicts such an example, where f's answer does not change when being provided with the counterfactual image. The cause of this failure is due to the difference between the original and counterfactual images is too small for f to change the prediction. This because the clashing constraints that $\mathcal{G}$ has to obey. For example, $\mathcal{G}$ is required to change the output of f but is at the same time restricted to apply as minimally as possible of changes. Another reason might be a wrong gaudiness of the attention map. Suppose the question-critical object accounts for a large portion of the image, or the question is about the image's background. In that case, $\mathcal{G}$ often only edits those parts on which the attention mechanism focuses. As a result, the relevant image region is not modified in its entirety and thus f cannot perceive a semantically meaningful change or its attention is shifted towards other regions of the question-critical object.

While $\mathcal{G}$ can generate semantically meaningful counterfactual images for a lot of color-based questions, it is not the case for shape-based questions as shown in Figure 37. While the VQA system predicts a different answer in both cases, the changes are not semantically meaningful from a human observer's perspective. In example 37a, the object's shape remains roughly unchanged, while the counterfactual generator slightly edits the sign's color. However, the task of $\mathcal{G}$ is accomplished, where the interpretation is that the VQA system is prone to any small change in the input to generate a different answer. Furthermore, f predicts an incorrect answer for both the original and the counterfactual image. As M indicates, the changes from the original image seem to be significant enough for f to shift its focus

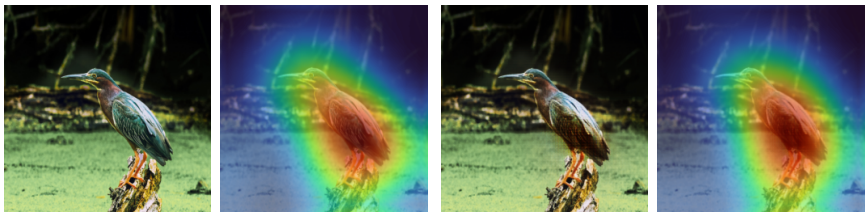
(a) Question: What color are the peppers in the bottom left corner?

f**'s answer**: yellow f**'s answer**: red

(b) Question: What color is the large central flower?

f**'s answer**: yellow f**'s answer**: orange

(c) Question: What color is the bird?

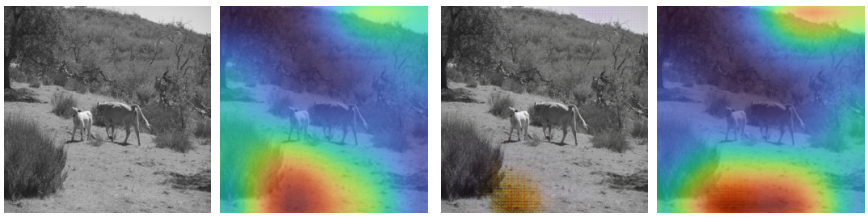


f**'s answer**: blue and white f**'s answer**: brown and white

(d) Question: What color is the dog?

f**'s answer**: brown f**'s answer**: brown

(e) Question: What is the color scheme of the picture?



f**'s answer**: black and white f**'s answer**: orange and white

Figure 36: Example outputs of $\mathcal{G}$ for color-based questions from the VQAv1 [19] validation set. Left: original image I and the corresponding attention map M. Right: Generated counterfactual image I' and the corresponding attention map M'.

(a) Question: What shape is the sign at the top of the post?

f**'s answer**: triangle f**'s answer**: square

(b) Question: What shape is the kite at the top left of the image?

f**'s answer**: triangle f**'s answer**: round

Figure 37: Example outputs of $\mathcal{G}$ for shape-based questions from the VQAv1 [19] validation set. Left: original image I and the corresponding attention map M. Right: Generated counterfactual image I' and the corresponding attention map M'.

slightly to the lower right portion of the sign when making an inference on the counterfactual image.

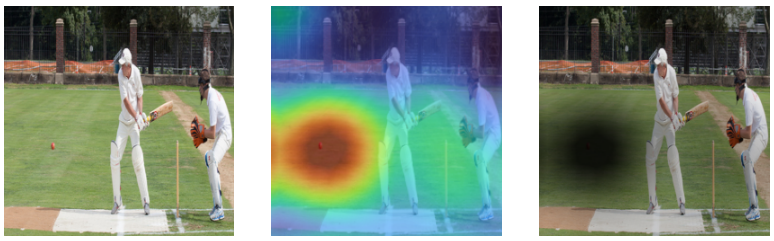
This shift seems to drive the model to change its prediction. Contrarily, the changes in example 37b are more dominant, where $\mathcal{G}$ produces an artifact covering the kite and a segment of the sky surrounding it. As M shows, f focuses on the same area in both the original and the counterfactual image, but the artifact seems to be the cause of the different answer. These two instances are exemplary for most of the counterfactual images for shape-based questions: the generator (i) applies only a few edits that are barely noticeable but they are more likely to change the answer of f which is the task of $\mathcal{G}$ or (ii) produces artifacts that are not semantically meaningful for a human observer.

Both the examples in Figure 36 and Figure 37 show that $\mathcal{G}$'s edits vary depending on the questions and answers. Since the attention maps pose a strong constraint for $\mathcal{G}$, its edits heavily depend on them. If M focuses on the question-critical object, such as in Figure 36c, $\mathcal{G}$ successfully modifies it. Contrarily, if M focuses only on a small portion of the object/region of interest (such as in Figure 36e), the language conditioning does not have the desired effect. In these cases, $\mathcal{G}$ fails to modify the areas relevant to the

question-answer pair sufficiently. Despite the failure to generate the counterfactual image, $\mathcal{G}$ provides an understandable interpretability to the behaviour and result of the VQA system for a particular pair (Image, Question).

Question-critical object (RQ2): One of the main aims of *COIN* is to edit only the question-critical object. This is controlled by the attention map M, where the generator $\mathcal{G}$ is restricted to apply changes predominantly in the regions on which M focuses. Figure 38 depicts, for three example samples, the original image I (Left), the question Q, the answer A given by MUTAN (f) and the interpolation of the map M with I (Center). The right image is the *background* obtained by computing $(\mathbf{1} - M) \odot I$, where $\mathbf{1}$ denotes all-ones matrix and $\odot$ is the element-wise multiplication.

(a) Question: What color is the ball? **MUTAN answer**: orange.



(b) Question: What color is the bus? **MUTAN answer**: red.

(c) Question: What color are the flags? **MUTAN answer**: Red and white.

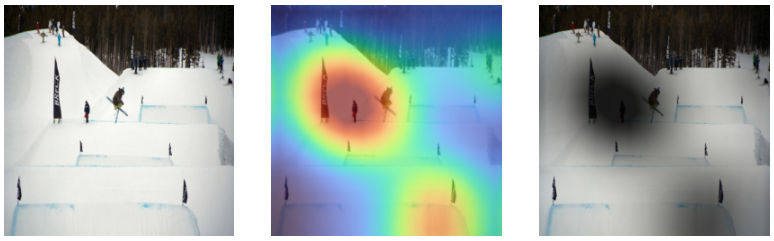


Figure 38: Example outputs of the Grad-CAM algorithm applied to MUTAN for color-based questions. Left: original image. Center: Interpolated attention map projected on the original image. Right: The background image.

In Figure 38a, the question is about the ball and as shown in the in-

terpolated map, f focuses specifically on the ball's region, which makes $\mathcal{G}$ restricted to make changes only on that region. When the region of interest is large and/or sparse such as in Figure 38b, f might not focus on the entire question-critical region but only a portion of it, which is sufficient to answer the question. Figure 38c indicates that f wrongly answered the question as *red and white* but the correct answer is clearly *black*. The obtained map M explains that f was focusing on a different region. Consequently, $\mathcal{G}$ applies the changes on the wrong region. The user can understand the behaviour of the model based on this generated image, which is an edit to the original one w.r.t a wrong region.

According to our user study, $\mathcal{G}$ applied the changes on the critical object in 50.1% of the samples. For 32.2% of the samples, the changes were applied on (1) completely different region, (2) the region of interest and other irrelevant regions or (3) a small part of the region of interest. For the remaining 17.7%, the users couldn't determine whether the changes were applied on the question-critical object or not. From these results, we can derive three main patterns of the attention mechanism:

1. If the object relevant to answering a question is relatively small compared to the rest of the image, the attention mechanism focuses on it completely in most cases. In other words, the computed intensities are higher for pixels belonging to the object than for the rest of the pixels. Under these circumstances, the generator can make larger changes to the entire object than to the rest of the image.

2. Contrarily, if the object is very large or MUTAN pays attention to the background, the projection usually focuses only on a part of it. Consequently, the information that the generator receives allows it to apply more significant changes to a segment of the object or the background than to the rest of it.

3. If MUTAN makes an incorrect prediction, this is often reflected by the projection not focusing on the question-critical object, but another element of the image, such as in Figure 38c.

In all these patterns, the applied changes of I w.r.t M is supposed to change the answer of f for the same question Q. This is because M is indicating where f is focusing to answer Q given I. However, when the VQA focuses on an irrelevant region in I, the applied changes might change the visual semantic of this irrelevant region such that when feeding f with I' and Q, f focuses on a different region than it did in I. This different region can also be the correct region.

Realism (RQ3): Generating realistic counterfactual images is very important to interpret the result of VQA systems to users. As shown in Figure 36

and Figure 37, the degree of realism varies depending on the necessary edit that changes f' answer and on the size difference of the question-critical region to the focused object. This is reflected also in the result of our user study that is demonstrated in Figure 39, where the users were presented only with the generated images and asked "*Does the picture look photoshopped (any noticeable edit or distortion)?*". As Figure 39 indicates, the answers of the users vary depending on the generated images. When presenting the corresponding original image together with the generated one and asking "*Which of the images is the original?*", the participants could correctly distinguish between the original and the generated one in $\sim$ 67.2% of the presented samples. In $\sim$ 12.9% of the images, the participants selected the generated image as the original one and in the rest of image ($\sim$ 19.9%), the participants couldn't decide which of the images is the original and which one is the generated. This result indicates that $\mathcal{G}$ could trick the human participants by generating counterfactual which look extremely realistic.

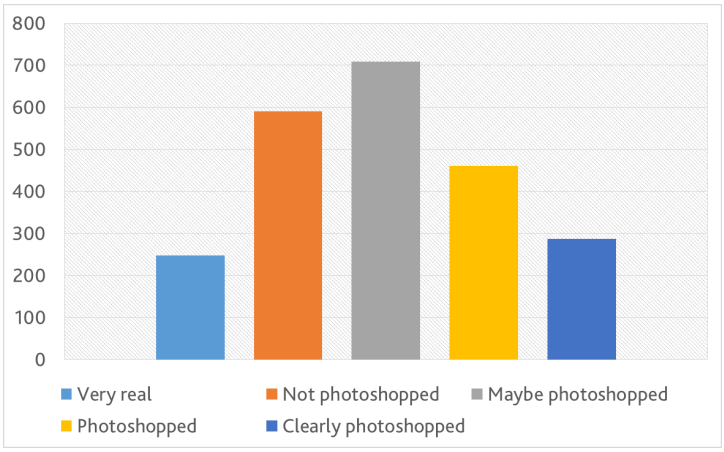


Figure 39: Frequency histogram of participants' answers to the question: "*Does the picture look photoshopped?*"

Minimality of Image edits: To evaluate $\mathcal{G}$'s performance on generating counterfactual images with minimum edits, we computed ℓ_1-norm across both the training and the validation set. This measures the magnitude of changes in the counterfactual image relative to the original one, where lower values indicate fewer changes.

Table 41 summarizes the results of this evaluation, where the mean (denoted μ) and standard deviation (denoted σ) values are calculated for different splits of both the training and the validation sets. The first three columns represent the values computed across the entire dataset and for color-based and shape-based questions. The remaining six columns contain the same computations for the portion of pairs of original and counterfactual images for which $\mathcal{G}$ predicts distinct or equal answers, respectively.

		Training Set		Validation Set	
		μ	σ	μ	σ
All		0.0175	**0.0039**	0.0175	0.0041
Color		**0.0174**	**0.0039**	**0.0174**	**0.004**
Shape		0.0177	0.0048	0.0208	0.0047
Same VQA Answers	ALL	0.0207	0.0040	0.0177	0.0041
	Color	**0.0176**	**0.0039**	**0.0176**	0.0041
	Shape	0.0212	0.0049	0.0212	0.0048
Different VQA Answers	ALL	0.0173	0.0039	**0.0173**	**0.0038**
	Color	**0.0172**	**0.0038**	**0.0173**	**0.0038**
	Shape	0.0195	0.0046	0.0198	0.0042

Table 41: Mean (μ) and standard deviation (σ) of the ℓ_1-norm computed across the training and validation set and split across categories

The results of Table 41 indicate that $\mathcal{G}$ applied fewer changes when it comes to color-based questions compared to shape-related questions. This observation applies to both the training and the validation set and across all splits. Moreover, overall, $\mathcal{G}$ applied fewer changes in those cases where f's predictions regarding the original and counterfactual image were distinct than if they were equal. This indicates that $\mathcal{G}$ changed the original image to the maximum possible level but without successfully changing f's answer.

9.5 Conclusion

In this paper, we introduced *COIN* , a GAN-based approach to interpret the output of VQA models by generating counterfactual images to drive the VQA model outputting different a different answer (***RQ1***). *COIN* is a modified implementation of LingUNet with incorporating a Grad-CAM-based attention mechanism that determines each pixel's importance regarding the VQA model's decision making process. With this, the counterfactual generator learns to apply modifications in an image predominantly to question-critical objects, while retaining the rest of the image (***RQ2***). The obtained results indicate that using an attention mechanism is an appropriate means to guide the modification process. Furthermore, the quality of the counterfactual images depended to a large extent on the attention maps. Extensive experiments on the challenging VQAv1 dataset have demonstrated that *COIN* achieves encouraging results for color questions by generating realistic counterfactual images (***RQ3***).

For future work, we will train *COIN* on a larger, more diverse dataset such as VQAv2 dataset, which contains multiple images per question rather than only a single one as in the VQAv1 dataset. Moreover, using an attention mechanism that focuses on the question-critical objects more accurately

could also significantly improve the interpretability capabilities of *COIN*. To this end, we plan to employ super-pixel segmentation to extract concepts (e.g. color, texture, or a group of similar segments) and uses the Shapley Value algorithm to determine each concept's contribution to a DNN's decision. The generator will then be trained to alter the most important concept(s) in an image rather than providing it an attention map. Moreover, replacing an entire instance of a concept rather than editing an image on a pixel-by-pixel level could pave the way for semantic changes even larger than altering shapes.

10 Paper 11: LLM Fine Tuning Optimisation

EMORL: Ensemble Multi-Objective Reinforcement Learning for Efficient and Flexible LLM Fine-Tuning

Lingxiao Kong, Cong Yang, Susanne Neufang, Oya Deniz Beyan, ***Zeyd Boukhers***(✉)

(DOI: 2025.sigdial-1.33)

Abstract Recent advances in reinforcement learning (RL) for large language model (LLM) fine-tuning show promise in addressing multi-objective tasks but still face significant challenges, including complex objective balancing, low training efficiency, poor scalability, and limited explainability. Leveraging ensemble learning principles, we introduce an Ensemble Multi-Objective RL (EMORL) framework that fine-tunes multiple models with individual objectives while optimizing their aggregation after the training to improve efficiency and flexibility. Our method is the first to aggregate the last hidden states of individual models, incorporating contextual information from multiple objectives. This approach is supported by a hierarchical grid search algorithm that identifies optimal weighted combinations. We evaluate EMORL on counselor reflection generation tasks, using text-scoring LLMs to evaluate the generations and provide rewards during RL fine-tuning. Through comprehensive experiments on the PAIR and Psych8k datasets, we demonstrate the advantages of EMORL against existing baselines: significantly lower and more stable training consumption ($17,529 \pm 1,650$ data points and $6,573 \pm 147.43$ seconds), improved scalability and explainability, and comparable performance across multiple objectives.

Keywords: *LLM; Multi Objective Optimization*

10.1 Introduction

Multi-objective optimisation for large language models (LLM) is a crucial research direction for natural language processing (NLP) tasks that need to fulfil diverse and often competing requirements [350, 290, 200]. Such tasks require models to simultaneously optimise for multiple evaluation criteria, balancing trade-offs between different objectives. This paper focuses on the counsellor reflection generation task, where an LLM reflects on users' prompts with empathy, marking the first step in therapeutic communication [266]. As exemplified in Figure 40, this task requires reflective responses that optimise multiple key objectives, such as reflection, empathy, and fluency in high-quality counselling interactions. We employ reinforcement learning (RL) to fine-tune pre-trained LLMs for these specific objectives [216, 270]. Conventional RL fine-tuning approaches combine multiple

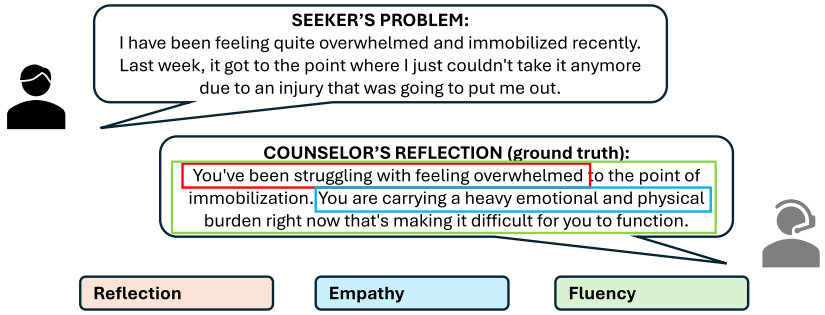


Figure 40: The counsellor reflection generation task requires creating reflective, empathetic, and fluent statements in response to the seeker's problem.

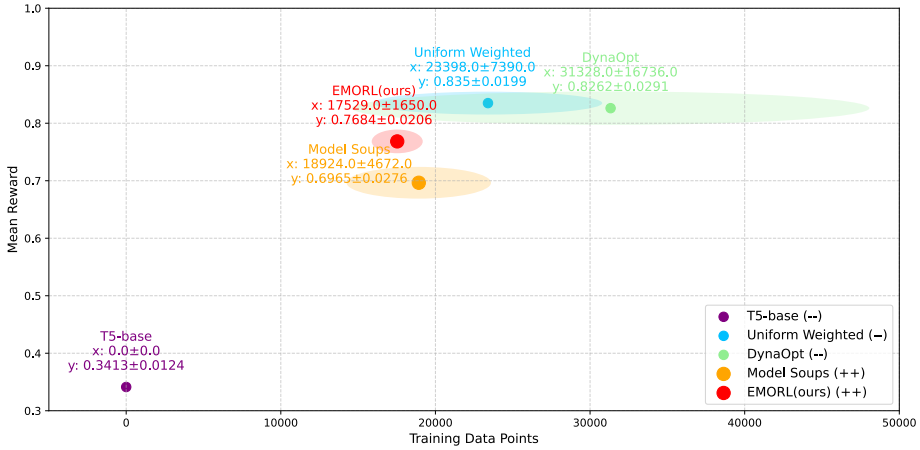


Figure 41: Comparison of training efficiency (x-axis) and mean reward (y-axis). Point size and "+/-" indicate scalability and explainability capabilities. EMORL achieves comparable mean rewards with lower data consumption while maintaining better scalability and explainability.

objectives into one reward function, enabling optimisation of various and conflicting objectives during training [264, 278, 84]. However, these approaches face significant challenges: maintaining training efficiency, ensuring scalability as objectives increase, and optimising appropriate weights for each objective while preserving result explainability [145, 101]. These challenges highlight the need for more efficient and flexible fine-tuning methods.

Ensemble learning offers a potential solution for this growing need by aggregating multiple trained models into a global model [348]. We propose a novel Ensemble Multi-Objective RL (**EMORL**) framework for LLM fine-tuning that distributes objectives to respective models, enabling training for individual objectives independently. The results demonstrate that models trained with single-objective reward functions converge significantly faster than those optimizing multiple objectives simultaneously. The framework incorporates a hidden states aggregation method coupled with a hier-

archical grid search algorithm during aggregation, which efficiently identifies optimal weights for combining these single-objective models. Our experiments demonstrate that the aggregated output achieves higher training efficiency while achieving performance comparable to models trained using single-policy methods, as shown in Figure 41. This framework is also highly scalable, allowing additional objectives to be incorporated as modular components. Additionally, the framework enhances explainability by providing insights into the relative importance of different objectives easily through evaluating different weighted combinations of objectives on test samples. The code for our framework and experiments is publicly available and can be found at `https://github.com/engineerkong/EMORL`.

Succinctly, our main contributions are as follows: (1) We introduce an ensemble multi-objective RL (**EMORL**) framework that separately trains and aggregates models in the counselor reflection generation task. (2) We develop an effective hidden-state level aggregation method and a hierarchical grid search algorithm for optimizing the weighted combination. (3) We demonstrate our framework's effectiveness through a comprehensive evaluation against multi-objective baselines, achieving comparable performance across multiple objectives while offering lower and more stable training resources, improved scalability and explainability.

10.2 Related Work

Prior work on RL for LLM fine-tuning has explored various approaches to balance multiple objectives. The most common approaches use a single policy to learn all objectives simultaneously, requiring careful objective weight selection and balancing within the reward function or loss function. Below, we analyze different techniques and their respective advantages and disadvantages. Additionally, we introduce ensemble learning methods, which combine multiple policies to create a meta-policy that addresses multiple objectives.

10.3 Single-policy

In the most conventional approaches, objectives are combined into one reward function or loss function using fixed weights λ_i for individual objectives, as shown in Equation (55), where the total reward R_{total} is summed up using respective rewards R_i.

$$R_{total} = \lambda_1 \cdot R_1 + ... + \lambda_n \cdot R_n \tag{55}$$

[399] explored RL for LLM fine-tuning using fixed weights to combine task-specific rewards with auxiliary objectives like fluency. However, selecting appropriate weights to balance objectives effectively is challenging, since it requires extensive trial-and-error to verify the effectiveness of weighted

combinations. [242] proposed AutoRL to automate the selection of optimal weights. However, it still requires training numerous candidate policies to trace back the optima, remaining computationally intensive.

Dynamic weighting methods adjust objective weights during training based on the model's performance, context, or external feedback, enabling a more flexible and adaptive balance between competing goals. Reward-driven approaches like [278]'s Multi-Armed Bandit and [287]'s Markov Chain strategies enable continuous adaptation based on received rewards. However, this approach requires these additional mechanisms to adjust weights, which increases computational complexity and introduces training instability. These limitations are evident in Figure 42, where the dynamic weighting method DynaOpt exhibits higher training costs and greater instability than the Uniform Weighted approach.

10.4 Meta-policy

To address these inefficiencies, meta-policy approaches use ensemble learning to reduce computational overhead and complexity. Ensemble learning comprises three main approaches: bagging (combining separately trained models), boosting (sequential training to improve upon previous models), and stacking (using a meta-learner to integrate outputs from diverse models, requiring additional meta-training) [323]. [362] demonstrated Model Soups that combines the parameters from multiple trained models using the concept of bagging. Their learned soup approach learns the weights for parameter-level aggregation and aggregates multiple models into a single model through gradient-based optimization. [316] combined the predicted tokens from models trained on individual objectives, namely logit-level aggregation, and leveraged Legendre transformation and f-divergence to optimize the performance of combined predictions for NLP tasks. [110] balanced the minimum throughput and standard deviation in load balancing by learning a meta-policy using distilled data from multiple trained models. [232] trained a meta-policy by stacking three types of feature embedding representations from different trained models and used a multi-head attention mechanism to non-linearly combine the features for scoring sentiment analysis.

These meta-policy approaches shift the objective balancing from the training phase to the model aggregation, making the multi-objective optimization more flexible. However, as demonstrated in Appendix A.1 and A.2, parameter-level aggregation and logit-level aggregation underperform in multi-objective optimization, particularly when the objectives differ significantly. This indicates that the application of ensemble learning in this multi-objective context remains underexplored. In light of this, we explore the novel hidden-state level aggregation by combining the contextual information from individual objectives to achieve efficient and flexible multi-objective LLM fine-tuning, while maintaining comparable performance.

10.5 Challenges

We analyze the challenges of multi-objective optimization in this section by comparing the training processes of models with individual objectives against multi-objective models using conventional single-policy methods. We tested five fine-tuning setups for counselor reflection generation to demonstrate these challenges, focusing on the objectives: reflection, empathy, and fluency. Three setups optimized single objectives separately, while two multi-objective approaches fine-tuned for all three objectives: (1) Uniform Weighted, assigning equal weights ($\frac{1}{3}$) to each objective, and (2) DynaOpt, dynamically adjusting weights using a multi-armed bandit algorithm. Experiments used 5 random seeds with 3 generation runs each, evaluating mean rewards, data consumption, and training time. Progress was tracked via Weights & Biases, plotting mean reward against data consumption.

As shown in Figure 42 and Table 44, our results highlight key differences between single-objective and conventional multi-objective fine-tuning in three aspects. First, **convergence speed**: single-objective models converged faster, with fluency models achieving the quickest convergence ($4,809$ data points, $1,629.19$ seconds). Multi-objective models were significantly slower, with Uniform Weighted requiring $23,398$ data points and $5,967.84$ seconds, and DynaOpt needing $31,328$ data points and $8,029.15$ seconds due to the overhead of dynamic weighting. Second, **process stability**: single-objective fine-tuning showed consistent convergence with minimal variation (± 335 data points, ± 104.40 seconds for reflection). Multi-objective models exhibited less stability, with Uniform Weighted varying by $\pm 7,390$ data points and $\pm 1,875.50$ seconds, and DynaOpt by $\pm 16,736$ data points and $\pm 4,365.64$ seconds, reflecting the complexity of balancing multiple objectives during training. Third, **performance metrics**: single-objective models achieved higher rewards (approaching 1.0 for reflection and empathy), while multi-objective models averaged below 0.85, indicating inherent performance trade-offs in optimizing multiple objectives simultaneously.

These observations suggest that integrating multiple objectives inherently presents convergence, stability, and performance challenges. A promising research direction emerges from this insight: **ensembling single-objective models to achieve multi-objective optimization**.

10.6 Methodology

As shown in Figure 43, the EMORL framework consists of three key stages. First, multiple models are independently fine-tuned for distinct individual objectives during training. Second, we employ a simple yet effective hierarchical grid search to explore optimal linear weighted combinations for aggregating these trained models at the hidden-state level. Finally, we use the optimal weights to aggregate the models at the inference stage, generat-

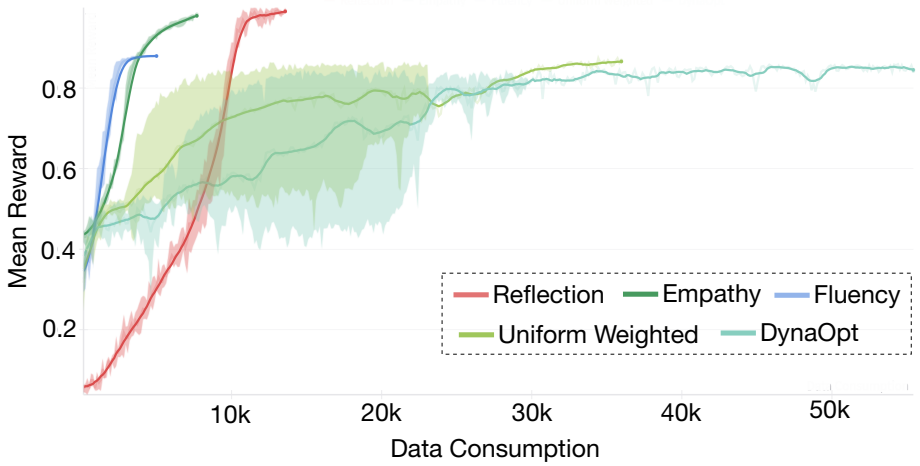


Figure 42: RL fine-tuning processes logs for 5 setups, highlighting single-objective models' advantages in convergence speed, process stability, and performance.

ing outputs that effectively integrate all objectives. This ensemble learning approach simplifies the complex multi-objective fine-tuning problem into an optimization problem for aggregation weights.

10.7 Hidden-State Level Aggregation

The decoder of an LLM generates hidden states that capture high-level features for contextual understanding, semantic representations, cross-attention patterns, and task-specific information [293]. Typically, the last hidden states are processed by a language model head to compute logits, representing token probabilities across the vocabulary list. Logits are then used to generate tokens via the arg max operation. In our approach, we aggregate the last hidden states from multiple objective-specific models to cohesively integrate their high-level features using a linear combination, as formulated in Equation (56).

In contrast to parameter-level and logit-level aggregation methods, whose experimental results we present in Appendix A.1 and A.2 respectively, our hidden-state level aggregation approach ensures more consistent text generation while effectively incorporating features from all objective-specific models. The weight coefficients in this linear combination determine each objective's relative contribution of contextual information to the final output.

$$\begin{aligned} \mathbf{f} &= \alpha \cdot \mathbf{f}_1 + \beta \cdot \mathbf{f}_2 + \gamma \cdot \mathbf{f}_3, \text{where } \alpha, \beta, \gamma \in [0, 1] \\ \mathbf{h}_t &= H_{LM}(\mathbf{f}) \in \mathbb{R}^{|V|} \\ \text{token}_t &= \underset{i \in \{1,2,\ldots,|V|\}}{\arg\max} \; \mathbf{h}_t[i] \end{aligned} \tag{56}$$

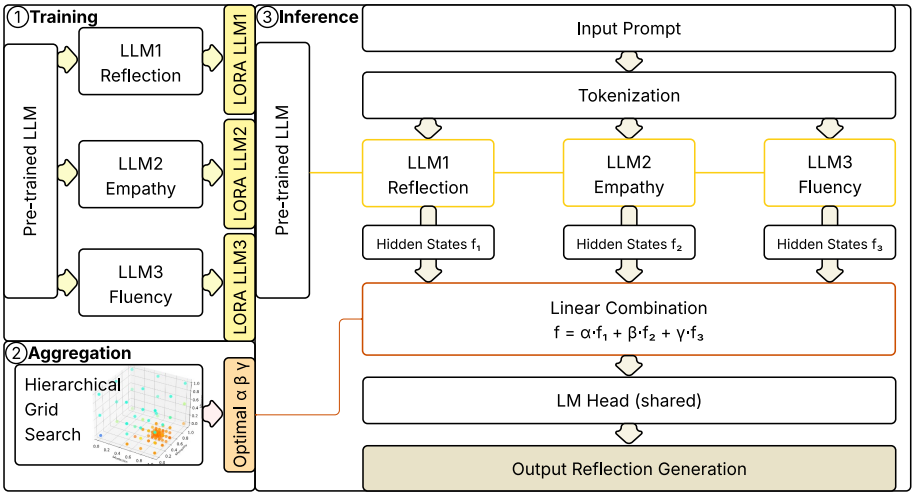


Figure 43: The EMORL framework illustrates a three-stage process: training, aggregation and inference.

Where $\mathbf{f}_i$ represents the hidden state from the i-th model, $\mathbf{f}$ is the linearly aggregated hidden representation, H_{LM} denotes the language model head transformation that projects the hidden state to the vocabulary space, $\mathbf{h}_t$ is the resulting logits vector of dimension $|V|$ (vocabulary size), and token_t is the selected token with maximum probability at timestep t. The coefficients α, β, and γ are weights constrained to the range $[0, 1]$, representing the independent contribution strength of each model.

10.8 Hierarchical Grid Search

To effectively and efficiently search for the optimal weights α, β, and γ, we experimented with various optimization methods. The principled Bayesian optimization approach described in the Appendix A.3 exhibited slow convergence due to sampling and populating numerous evaluations, often trapped in local optima. Through analyzing performance distributions, we observed that the optima consistently appeared within higher-performance regions, leading us to explore the grid search methods.

We propose a hierarchical grid search algorithm that incorporates binary search concepts to reduce the computational complexity inherent in standard grid search [39, 121]. The developed hierarchical grid search achieves a computational complexity of $O(3^d \cdot \log_2 \frac{1}{N})$ compared to grid search's $O(\frac{1}{N}^d)$, where d represents the objectives and N the precision level. The complexity comparison among these methods is shown in Figure 48. With three objectives and a precision level of 0.03125, the required number of evaluations is reduced to 135, compared to $32,768$ in standard grid search.

As detailed in Algorithm 2, our hierarchical approach first divides each search axis for individual objectives into 3 parts, creating 3^d initial grid

points. We then evaluate the generation performance at these points and identify the most promising region by finding the 2^d cube with the highest total performance score. This region becomes the next search space, and we iterate these processes of grid generation and space refinement. The algorithm progressively focuses on smaller, more promising regions, which proves particularly effective for this aggregation case.

Algorithm 2 Hierarchical Grid Search

Require: objective function f, number of components N, iterations I, initial bounds $B_0 = [(0, 1)]^N$
Ensure: Best point p^*, Best score s^*

$p^* \leftarrow$ null
$s^* \leftarrow -\infty$
$B_{\text{current}} \leftarrow B_0$
for iter $= 1$ **to** I **do**
    grid_points $\leftarrow$ GenerateGrid(B_{current})
    results $\leftarrow \{\}$
    for all point $p \in$ grid_points **do**
        results$[p] \leftarrow f(p)$
    end for
    $p_{\text{current}} \leftarrow \arg\max$(results)
    if results$[p_{\text{current}}] > s^*$ **then**
        $s^* \leftarrow$ results$[p_{\text{current}}]$
        $p^* \leftarrow p_{\text{current}}$
    end if
    region $\leftarrow$ FindBestRegion(results)
    $B_{\text{current}} \leftarrow$ ComputeBounds(region)
end for
return p^*, s^*

10.9 Experiments

10.10 Models

We employ **T5-base**[50] (220M parameters, Encoder-Decoder architecture) as the pre-trained model in this paper. [278] demonstrated T5's effectiveness for understanding and generating text in counseling tasks, making it suitable for our experiments compared with existing baselines. We utilize Self-Critical Sequence Training (**SCST**) as the RL algorithm, which generates candidate outputs and computes their mean reward as a baseline, thereby encouraging outputs to exceed this baseline performance [202]. This approach integrates KL-divergence in the loss function to constrain the fine-tuned model from diverging too far from the reference pre-trained model.

We fine-tune models for reflection, empathy, and fluency objectives across

[50] https://huggingface.co/google-t5/t5-base

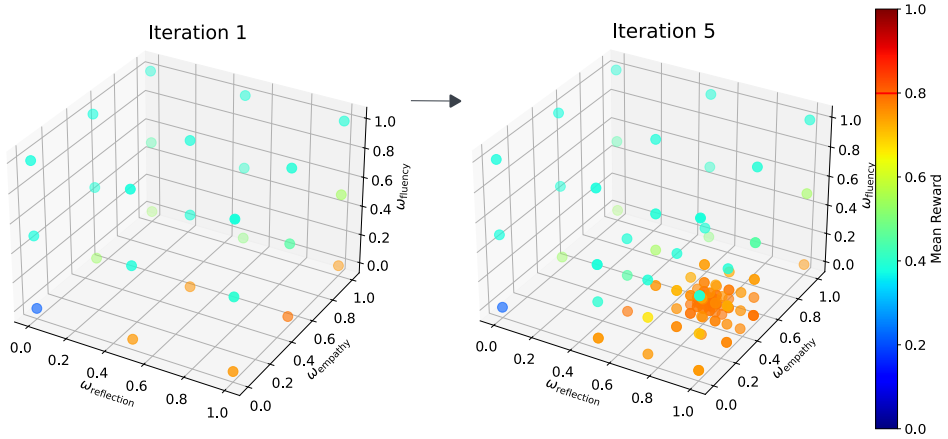


Figure 44: The visualization illustrates the hierarchical grid search process, showing the transition from broad search spaces to more refined spaces, where optimal weight combinations are identified. The red line on the color map indicates the maximum mean reward achieved during the search.

5 random seeds, with 3 generation runs per seed, a training batch size of 16, and up to 10,000 steps while implementing early stopping at model convergence. We also employ LoRA [161] to efficiently manage parameter updates by representing them via low-rank matrices and a scaling factor, as illustrated in Equation (58). This approach allows us to load the pre-trained model only once during inference and apply the LoRA parameters to update the pre-trained model toward each objective, significantly reducing the computational burden.

For evaluation, we compared EMORL models against four baselines: T5-base, Uniform Weighted, DynaOpt [278], and Model Soups [362]. All experiments were conducted on a Tesla V100 GPU with 32GB memory, 8 CPU cores, and 40GB system memory, with detailed consumption reported in Table 42.

10.11 Datasets

Counselor reflection generation task is a single-turn task to generate therapist-like reflective statements that accurately paraphrase and validate a client's expressed thoughts and feelings. It reflects the seeker's problem and establishes the first step for therapeutic communication. The responses are expected to be reflective, empathic to the problem, and maintain fluency, as depicted in Figure **??**.

The **PAIR**[51] dataset is our primary dataset, split into a ratio of 80%, 10%, and 10%, for training, aggregation, and inference, respectively [238]. It

[51]https://lit.eecs.umich.edu/downloads.html

contains 2,544 single-turn client-counselor exchanges, covering topics ranging from mental health to lifestyle concerns like diet, exercise, and personal development. To assess the models robustly, we also conduct evaluations on the **Psych8k**[52] dataset, sampling 10% of its 8,187 conversation pairs for inference. This dataset focuses on mental health interactions, including anxiety, depression, relationship issues, and stress management. It is widely used for training and evaluating LLMs in mental health counseling, and we leverage it here for generating reflective statements.

10.12 Metrics

We evaluate comprehensive metrics beyond performance, focusing on six key aspects: (1) **Diversity-2** measures linguistic diversity; (2) **Edit rate** quantifies the avoidance of verbatim repetition; (3) **Data consumption** tracks the cumulative number of training samples processed; (4) **Time consumption** records the wall-clock time for each training iteration; (5) **Scalability** assesses the model's ability and flexibility to incorporate additional objectives; (6) **Explainability** examines the interpretability of how each objective contributes to the final output.

For performance metrics, we employ specific LLMs to score three objectives on a scale from 0.0 to 1.0: (1) **Reflection** is assessed by the "roberta-base[53]" with checkpoints from [278], which evaluates the relevance and contextual appropriateness. (2) **Empathy** is measured by the "bert-empathy[54]", which gauges emotional resonance and understanding. (3) **Fluency** is evaluated using "gpt2[55]" by computing the inverse of perplexity, ensuring linguistic smoothness.

We conducted human evaluation of 640 generations sampled from five models (T5-base, Uniform Weighted, DynaOpt, Model Soups and EMORL) across two datasets (PAIR and Psych8k). Two mental health experts independently rated each response on three performance metrics using a 3-point scale, normalized to 0.0-1.0: (1) **Reflection**: 0 (no reflection), 1 (simple mirroring), 2 (complex interpretation); (2) **Empathy**: 0 (no emotional awareness), 1 (basic understanding), 2 (deep emotional resonance); (3) **Fluency**: 0 (poor coherence), 1 (clear but awkward), 2 (natural and clear). The assessment instruction is detailed in Appendix A.4.

10.13 Results Analysis

Hierarchical Grid Visualization Demonstrates Explainable Results. Figure 44 illustrates the hierarchical grid search process: in iteration 1 (left

[52]https://huggingface.co/datasets/EmoCareAI/Psych8k
[53]https://huggingface.co/FacebookAI/roberta-base
[54]https://huggingface.co/MoaazZaki/bert-empathy
[55]https://huggingface.co/openai-community/gpt2

Table 42: The comprehensive metrics highlight measures beyond performance. The results demonstrate that our EMORL framework offers advantages in generation diversity, low training consumption, enhanced scalability, and improved explainability, outperforming other methods.

	T5-base	Uniform Weighted	DynaOpt	Model Soups	EMORL (ours)
Diversity-2 (↑)	0.8851±0.0056	0.3561±0.0837	0.3621±0.0951	0.4327±0.0932	**0.6516**±0.0524
Edit Rate (↑)	0.8087±0.0127	0.8870±0.0247	**0.8929**±0.0246	0.8672±0.0326	0.8734±0.0240
Data Consumption (↓)		23398±7390	31328±16736	18924±4672	**17529**±1650
Time Consumption (↓)		5967.84±1875.50	8029.15±4365.64	**5823**±1262.24	6573±147.43
Scalability		-	-	+	+
Explainability		-	-	+	+

subplot), we evaluated $3 \times 3 \times 3$ weighted combinations for reflection, empathy, and fluency, with color mapping indicating mean reward values. The $2 \times 2 \times 2$ area with best total performance was then used to refine the search space for subsequent iterations. The right subplot of Figure 44 shows iteration 5, where the search converged to a more precise and refined space: $(0.75, 0.8125)$, $(0.4375, 0.5)$ and $(0.0, 0.0625)$ for the aggregation weights of reflection, empathy, and fluency models respectively. This progressive refinement allows for the precise identification of the optimal weighted combination.

The visualization reveals significant insights regarding each objective's contribution to overall performance. The aggregation benefits (orange points) from higher weights of the reflection model. Empathy delivers optimal overall performance with moderate weights, while extreme weights reduce the mean reward. The fluency model demonstrates a negative effect (cyan points) on other objectives when assigned high weights, with lower weights facilitating better integration with the other objectives. This visualization can be readily acquired during aggregation by evaluating weighted combinations on batches of test data. Although results vary across experiments with different sampled models, this approach underscores the interpretability and explainability of EMORL.

EMORL Shows Promise Across Multiple Evaluation Metrics. As shown in Table 42, EMORL achieves the highest diversity-2 score among fine-tuned models, averaging above 0.65, compared to $0.35 - 0.43$ for other fine-tuned models. This highlights EMORL's ability to generate diverse responses, which is achieved by aggregating hidden states and delegating the token generation to the language model head. EMORL has an edit rate of 0.8734, which is very close to other models, indicating that it avoids verbatim repetition.

EMORL demonstrates superior efficiency in resource utilization, consuming approximately $17,529$ data points and $6,573$ seconds of training time. Due to its ensembled architecture, the total resource consumption T_{total} is determined by:

$$T_{total} = \max_{i \in \{1,2,3\}} \{T_{train}(obj_i)\} + T_{agg} \tag{57}$$

This parallel process enables over $0.5\times$ faster training compared to single-policy methods. While another meta-policy method, Model Soups, employs gradient-based algorithms (learned soup) that often struggle with local optima, leading to increased consumption. However, EMORL employs hidden-state level aggregation and requires token-by-token generation, which interrupts the sequential generation process of transformers, slightly increasing time consumption. EMORL exhibits greater stability, with variations of only $1,650$ data points and 147.43 seconds, significantly lower than DynaOpt's

Table 43: The performance metrics are evaluated automatically and through human assessment on the PAIR and Psych8k datasets. The human-evaluated scores are averaged across 640 samples from both datasets. The overall results demonstrate that our EMORL method achieves performance comparable to other methods.

		Reflection (↑)	Empathy (↑)	Fluency (↑)
PAIR	T5-base	0.0418±0.0108	0.4648±0.0160	0.4849±0.0185
	Uniform Weighted	**0.9616**±0.0212	0.8078±0.0251	**0.7498**±0.0176
	DynaOpt	0.9349±0.0234	**0.8141**±0.0329	0.7271±0.0300
	Model Soups	0.9204±0.0315	0.7418±0.0264	0.4324±0.0186
	EMORL (ours)	0.9406±0.0406	0.7766±0.0178	0.6548±0.0113
Psych8k	T5-base	0.0968±0.0099	0.3198±0.0129	0.6397±0.0062
	Uniform Weighted	0.9694±0.0066	0.7317±0.0314	**0.7897**±0.0173
	DynaOpt	0.9755±0.0148	**0.7330**±0.0487	0.7725±0.0247
	Model Soups	0.9518±0.0126	0.6722±0.0235	0.4602±0.0162
	EMORL (ours)	**0.9784**±0.0164	0.6838±0.0268	0.7462±0.0108
Human	T5-base	0.2618	0.2563	0.6875
	Uniform Weighted	0.5074	0.4563	**0.4438**
	DynaOpt	**0.5608**	**0.5473**	0.3118
	Model Soups	0.5178	0.5122	0.2490
	EMORL (ours)	0.5308	0.4858	0.3758

variations of 16,736 data points and 4,365.64 seconds. This stability is attributed to EMORL's consistent single-objective training and uniform optimization resource consumption facilitated by hierarchical grid search. These advantages position EMORL as an efficient and stable fine-tuning approach for multi-objective tasks.

EMORL is a flexible fine-tuning approach that achieves both scalability and explainability. Its scalability is demonstrated when adding new objectives: instead of retraining existing models, EMORL only trains the new single-objective model and optimizes the weights through the aggregation phase. For explainability, EMORL provides clear insights through weighted combinations and hidden-state level aggregation patterns. Figure 44 reflects varying contributions: reflection approximately 0.8, empathy around 0.5, and fluency about 0.05. Adjusting one objective's weight enhances its performance while affecting other objectives. This trade-off is explainably illustrated in the visualization, which only requires testing on batches of test data. In contrast, conventional single-policy methods require entire retraining to incorporate new objectives and rely on extensive trial-and-error to determine each objective's importance.

EMORL Delivers Comparable Performance in Performance Metrics. We evaluated our EMORL method on the PAIR and Psych8k datasets for reflection, empathy, and fluency metrics. Although EMORL does not achieve the highest scores, it maintains performance comparable to the conventional single-policy models and significantly outperforms the Model Soups

models.

On the PAIR dataset, EMORL achieves an average score of 0.7907, outperforming parameter-level aggregation Model Soups (0.6982) and performing comparably to the single-policy Uniform Weighted (0.8397) and DynaOpt (0.8254) methods. Notably, model performance varies slightly between the two datasets. On Psych8k, EMORL achieves the mean reward of 0.8082, with its reflection score reaching the highest (0.9784) among all models. Compared to Model Soups, EMORL shows a substantial improvement of 0.25 in fluency scores, demonstrating the effectiveness of hidden-state aggregation.

Human evaluation scores, averaged across PAIR and Psych8k, are generally lower than automated metrics but show consistent trends. These evaluations support our findings: all fine-tuned models demonstrate improvements in reflection and empathy but exhibit a slight decline in fluency. EMORL achieves the second-highest scores in reflection (0.5308), empathy (0.4858), and fluency (0.3758) among fine-tuned models, demonstrating balanced performance across all metrics and underscoring its potential as an effective fine-tuning method.

The sample generations are shown in Figure 45. EMORL improves reflection and empathy by employing second-person speech ("you"), introducing new perspectives ("job", "office"), and crafting understandable and empathetic statements ("feeling") in response to prompts. EMORL can paraphrase prompts by reflecting on the seeker's problem and aligning well with the desired objectives for generations, highlighting its effectiveness.

10.14 Discussion and Conclusion

To conclude, our study addresses the challenges of multi-objective optimization in LLM fine-tuning. We identify the key limitations of convergence speed, process stability, and performance metrics in conventional single-policy approaches, where multiple linguistic objectives are combined into one reward function. To address these limitations, we propose EMORL, a novel meta-policy framework using ensemble learning to aggregate diverse models trained on individual objectives. The results demonstrate that EMORL achieves greater diversity, efficiency, scalability, and explainability while maintaining performance comparable to existing methods in counselor generation tasks.

Our approach is the first to aggregate hidden states to incorporate multiple objectives in NLP. Unlike classification or regression models, which can simply concatenate results from each model to collaboratively decide on the final output, LLMs produce complete sequential generations where fluency cannot be separated or neglected. The hidden states represent rich contextual information, which proves more valuable than incorporating parameters or logits. By using the hidden-state level aggregation, EMORL exceeds the

performance limitations of other meta-policy methods. EMORL also offers unique advantages over conventional single-policy methods, including improved training efficiency, extensive scalability, and enhanced explainability due to the parallel training properties, making it more flexible and interpretable than conventional single-policy methods. We investigated the nature of the multi-objective optimization problem and discovered that optima consistently appear within higher-performance regions. This insight led us to design a simple yet effective hierarchical grid search algorithm that requires fewer evaluations to find the globally optimal weights.

Our approach is both task- and model-agnostic. It's compatible with all transformer-based LLMs, as these architectures maintain hidden states that represent contextual information encompassing multiple objectives in NLP. This study demonstrates the potential of ensemble learning to advance current RL training paradigms and points to a promising novel direction for efficient and flexible multi-objective optimization in future applications.

Limitations

Our study has advanced a new paradigm of multi-objective optimization for LLM fine-tuning but faces several limitations that suggest directions for future research. First, the current implementation focuses on single-turn generation, which fails to capture the dynamics of counseling conversations. The RL interaction is limited to one-time evaluations without dialogue history, and reflections are generated based solely on prompts, not fully leveraging RL's potential for complex interactions. Future work should explore multi-turn conversation tasks, potentially incorporating dynamic weighting of model aggregation across dialogue turns.

Second, our study employs small-scale encoder-decoder LLMs, which may not achieve application-level performance. As shown in Figure 45, while EMORL generates the reflections addressing new empathic perspectives compared to the pre-trained model, the overall quality remains limited. This indicates that pre-trained model constraints affect generation quality, despite improvements in targeted behaviors. Future research should implement EMORL on larger models with billions of parameters to enhance performance and capabilities.

Finally, challenges remain in effectiveness and efficiency. While EMORL achieves comparable results across objectives, improving performance to surpass conventional RL fine-tuning remains a key challenge, which could potentially be addressed through non-linear combination approaches. Additionally, hidden-state level aggregation requires token-by-token generation, impacting the sequential generation process and slightly increasing time consumption. Future work should explore advanced aggregation methods to enhance computational efficiency and output quality while preserving the benefits of ensemble learning.

Potential Risks

We suggest that our models are not advocated for deployment in clinical or mental health settings. This is because human understanding and communication are indispensable in these domains, and the behavior of language models remains incompletely explored. Instead, we propose that our method and models be utilized for methodological research.

Ethical Considerations

The PAIR and Psych8K datasets used in our study are either open-source or licensed under CC-BY-NC. These datasets include one-turn motivational interviewing conversations as well as mental health interactions between counselors and patients. We ensured that the source datasets processed the dialogues to redact any personally identifiable information. Generative AI was employed solely to assist with bug fixing and grammatical error correction. All other work presented in this paper was conducted entirely by us.

Appendix A:

Table 44: Comparison of single-objective and multi-objective fine-tuning in addressing the challenges.

	Mean Reward (↑)	Data Consumption (↓)	Time Consumption (↓)
Reflection	0.9967 ± 0.0028	13209 ± 335	4175.05 ± 104.40
Empathy	0.9935 ± 0.0037	7136 ± 474	2232.18 ± 82.62
Fluency	0.8803 ± 0.0003	4809 ± 178	1629.19 ± 58.03
Uniform Weighted	0.8489 ± 0.0172	23398 ± 7390	5967.84 ± 1875.50
DynaOpt	0.8318 ± 0.0076	31328 ± 16736	8029.15 ± 4365.64

A.1 Parameter-level aggregation

We investigated parameter-level aggregation of local models using LoRA updates, following Equation (58), where the final parameter matrix θ is constructed by adding weighted low-rank adaptations to the initial pre-trained parameters θ_0. Each adaptation consists of a pair of matrices B_i and A_i

Table 45: Demonstration of EMORL's best-performing weight combinations for 5 model pairs in the experiments, along with their average combination.

	Rewards	Reflection	Empathy	Fluency
A_1	**0.7936**	0.9375	0.71875	0.0625
A_2	**0.7960**	1.0	0.625	0.125
A_3	**0.8092**	1.0	0.625	0.125
A_4	**0.7942**	1.0	0.5	0.0625
A_5	**0.8093**	0.78125	0.5	0.0625
A	**0.8005**	0.94375	0.59365	0.0875

Table 46: PAIR and Psych8k datasets statistics.

statistics	PAIR dataset	Psych8k dataset
# of Exchange Pairs	2,544	8,187
Avg # of Words	32.39	45.18

Prompt: I have been feeling quite overwhelmed and immobilized recently. Last week, it got to the point where I just couldn't take it anymore due to an injury that was going to put me out.

(T5-base) Reflection: I feel very immobilized. I need to get back to work a bit, as I have been feeling an amount of a lot lately.

(Uniform Weighted) Reflection: You know you want to take anymore due to an injury that was going to put you out. You know you want to take your injury further.

(DynaOpt) Reflection: That's the fact that you feel quite overwhelmed and immobilized recently, you know that you need an injury that was going to put you out.

(Model Soups) Reflection: So overwhelmed and feeling that you can't take it anymore feeling that you are feeling so overwhelmed and immobilized.

(EMORL) Reflection: You are feeling overwhelmed and immobilized by the feeling of being out of the office with your injury that you were going to put you out of the job.

Figure 45: Sampled generations of different models on the counselor reflection generation task.

whose product forms a low-rank update, scaled by a scaling factor α_i, thus avoiding high-rank parameter updates. It focuses on optimizing the weights w_i to effectively incorporate each objective in the aggregation process.

$$\theta = \theta_0 + \sum_{i=1}^{n} (B_i A_i) \alpha_i w_i \tag{58}$$

Model Soups, a prominent ensemble learning method, utilizes this parameter-level aggregation strategy for optimizing hyperparameter configurations. However, when applied to multi-objective optimization, this approach yielded suboptimal results. As illustrated in Figure 46, the overall mean reward achieved only 0.6982, significantly lower than our hidden-state level aggregation method. The fluency metric performed particularly poorly, reaching merely 0.4324. This underperformance likely stems from the fundamental incompatibility between diverse model objectives at the parameter-level, ultimately failing to effectively combine multiple optimization objectives simultaneously.

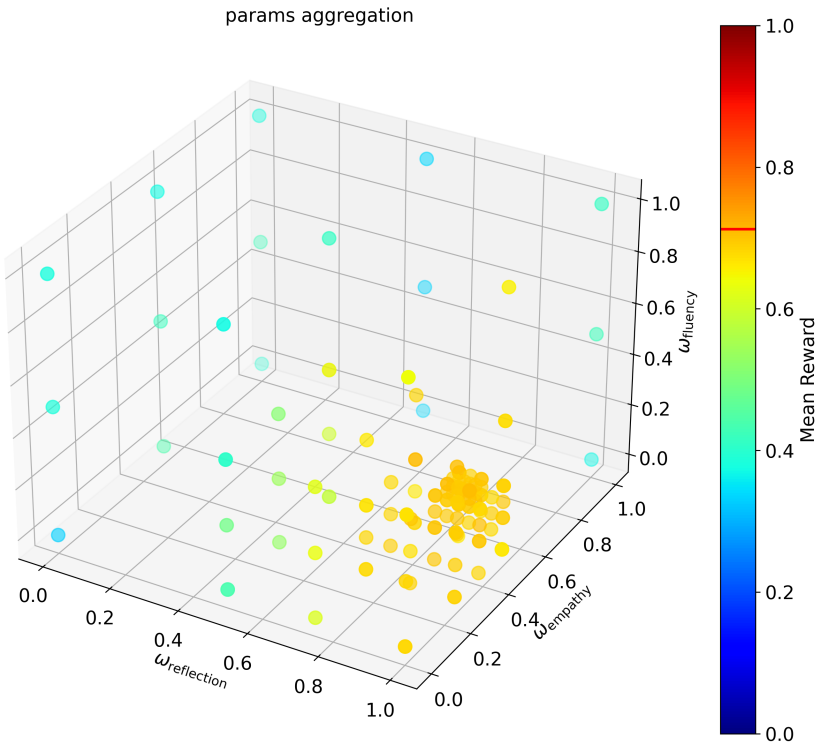


Figure 46: Parameter-level aggregation results.

A.2 Logit-level aggregation

We further explored logit-level aggregation as an alternative approach, which is grounded in the work of [316]. This method aggregates the logits, which represent token probabilities across the vocabulary and directly influence token generation. As illustrated in Figure 47, this approach performed even worse for combining multiple objectives, achieving a maximum mean reward of only 0.5934, with the fluency metric scoring a mere 0.1575. This poor performance can be attributed to the naive combination of vocabulary probabilities, where predicted token distributions from different local models are simply merged. This process severely impacts fluency, as the resulting text lacks coherence when calculated from disconnected probability distributions. In contrast, our last hidden states aggregation proves more effective by preserving and combining high-level contextual representations during generation, maintaining semantic consistency while still incorporating multiple objectives.

A.3 Optimization Algorithms

For a d-dimensional standard grid search with precision level N, the computational complexity is $O(\frac{1}{N}^d)$. This follows directly from evaluating the

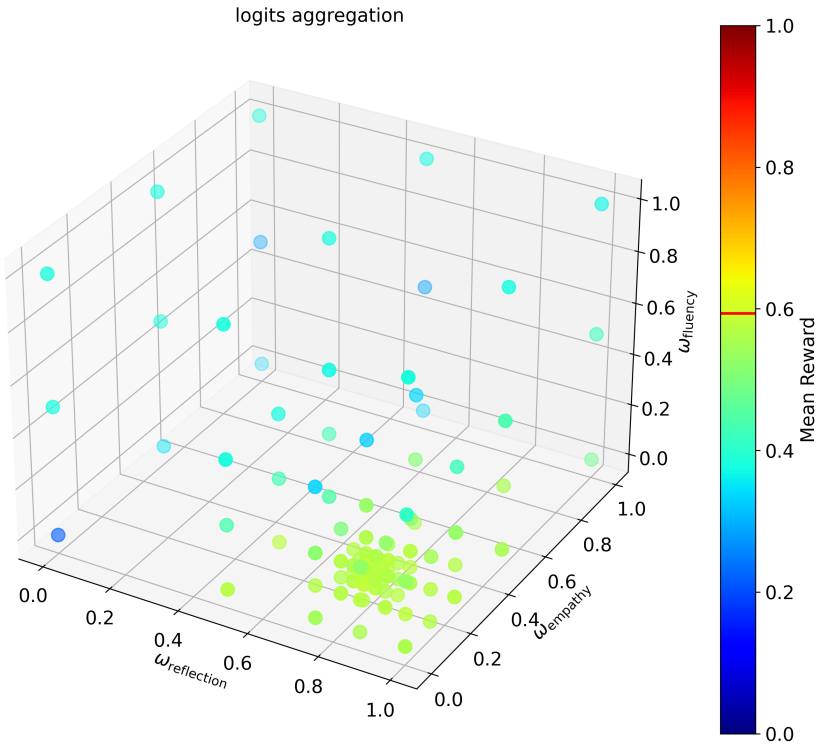


Figure 47: Logit-level aggregation results.

function at $\frac{1}{N}$ grid points in each dimension. In contrast, hierarchical grid search achieves $O(3^d \cdot log_2 \frac{1}{N})$ complexity by employing a hierarchical strategy. Starting with a coarse grid, the algorithm progressively refines only promising regions, halving the grid spacing at each level. To reach a final precision of N, it requires $log_2 \frac{1}{N}$ refinement levels. At each level, we evaluate 3^d points per promising region. While still exponential in dimension, the logarithmic dependence on precision offers substantial computational savings for fine-grained searches. A computational complexity comparison among these methods is shown in Figure 48, where the cost of Bayesian optimization is derived from our experimental estimates.

Bayesian optimization is a sequential strategy for optimizing black-box functions that are expensive to evaluate. Its process consists of building a probabilistic model (e.g., Gaussian Process) of the objective function based on previous observations, using this model to construct an acquisition function that determines where to sample next, evaluating the true objective function at this new point, updating the probabilistic model with this new observation, and repeating until convergence.

We applied Bayesian optimization to our weights optimization problem by allowing it to select appropriate weighted combinations and evaluate their generation performance. We conducted the optimization experiment across

several runs. The results showed that in approximately $\mathbf{1034 \pm 423}$ evaluations, Bayesian optimization could identify weights similar to those obtained through hierarchical grid search. Although the weights are continuous without specific precision levels, Bayesian optimization required significantly more computational resources compared to our hierarchical grid search, which evaluated only **135** combinations with a precision level of 0.03125. Furthermore, the stochastic nature of Bayesian optimization led to considerable instability in the optimization process.

We analyze the reason behind why hierarchical grid search is much more effective in this weights optimization problem. The hierarchical approach succeeds primarily because it exploits the structure of the weight space by systematically narrowing down promising regions. Rather than treating the objective function as a complete black box like Bayesian optimization does, hierarchical grid search leverages our prior knowledge that optimal weights likely exist within certain bounded regions. This allows for efficient pruning of unpromising areas early in the search process. Additionally, the deterministic nature of grid search provides consistent, reproducible results, eliminating the variability introduced by the stochastic nature of Bayesian methods. The precision level we implemented (0.03125) also worked well for this specific application, as this granularity proved sufficient for practical performance while dramatically reducing the search space compared to the continuous optimization attempted by Bayesian methods. Furthermore, the computational overhead of maintaining and updating probabilistic models in Bayesian optimization becomes significant when the objective function itself is relatively inexpensive to evaluate, making the more straightforward grid-based approach more efficient in this task.

A.4 Evaluation Instruction

The human evaluation is supported by two annotators, one is from China, and the other is from Germany. The evaluation, based on their cross-cultural understanding, supports the robust human-annotated results. This evaluation instruction is derived from the original instruction presented in the work [278]. When evaluating responses, choose the most appropriate score (0, 1, or 2) based on these criteria. Responses may vary in complexity, and the judgment should be guided by the degree to which they reflect upon the client's prompt.

Reflection: 0 (Non-Reflection), 1(Simple Reflection), or 2 (Complex Reflection). Non-Reflection (0): A response is considered a non-reflection when it does not engage with the client's input or the task at hand. It may be off-topic, irrelevant, or simply fail to address the client's query. Simple Reflection (1): A response is categorized as a simple reflection when it acknowledges the client's input or question without adding substantial depth or insight. It might repeat or rephrase the client's words, showing under-

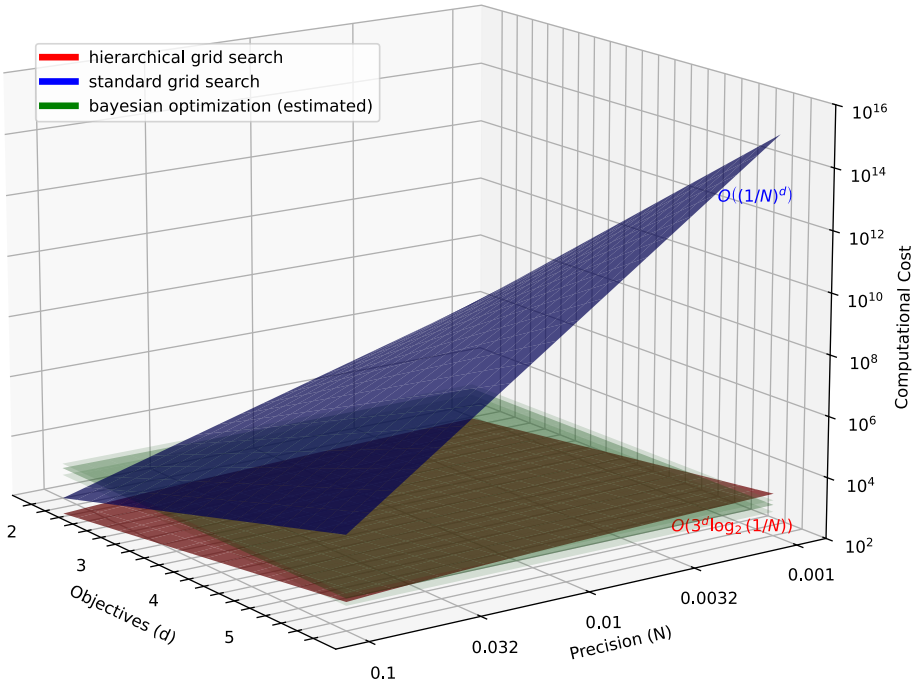


Figure 48: Computational complexity comparison between hierarchical grid search, standard grid search, and Bayesian optimization across varying numbers of objectives and precision levels.

standing but not extending the conversation significantly. Simple reflections demonstrate basic engagement with the client'squery. Complex Reflection (2): A response is identified as a complex reflection when it goes beyond mere acknowledgment and engages deeply with the client's input or question. It demonstrates an understanding of the client's thoughts, feelings, or concerns and provides a thoughtful, insightful, or elaborate response. Complex reflections contribute to the conversation by expanding upon the client's ideas or by offering new perspectives.

Empathy: 0 (Non-Empathetic), 1 (Basic Empathy), or 2 (Advanced Empathy). Non-Empathetic (0): A response that shows no recognition or acknowledgment of the person's emotional state or perspective. E.g. Dismiss or invalidate feelings. Change the subject without addressing emotions. Offer purely factual or technical responses when emotional support is needed. Show complete misalignment with the person's emotional state Basic Empathy (1): A response that demonstrates fundamental recognition of emotions and attempts to understand the person's perspective. E.g. Acknowledge obvious or stated emotions. Use basic emotional labelling ("That must be hard"). Mirror the person's expressed feelings. Show surface-level understanding without deeper exploration. Offer general supportive statements. Advanced Empathy (2): A response that shows deep emotional attunement and sophisticated understanding of the person's experience. Connect dif-

ferent aspects of the person's experience and recognize nuanced emotional states. Demonstrate understanding of the broader context and implications. Show genuine emotional resonance while maintaining appropriate boundaries. Help the person gain new insights into their emotional experience.

Fluency: Assess the linguistic naturalness and smoothness of the counsellor's responses. Responses are rated on a scale from 0 to 2, where 0 indicates responses that lack fluency, 1 signifies somewhat fluent responses, and 2 represents responses that are highly fluent and natural in their expression. Fluent counsellor responses should convey information in a clear and easily understandable manner, ensuring effective communication.

11 Paper 12: LLM Compression

Exploring the Limits of Model Compression in LLMs: A Knowledge Distillation Study on QA Tasks

Joyeeta Datta, Niclas Doll, Qusai Ramadan, ***Zeyd Boukhers***(✉)

(DOI: 2025.sigdial-1.39)

Abstract Large Language Models (LLMs) have demonstrated outstanding performance across a range of NLP tasks; however, their computational demands hinder their deployment in real-world, resource-constrained environments. This work investigates the extent to which LLMs can be compressed using Knowledge Distillation (KD) while maintaining strong performance on Question Answering (QA) tasks. We evaluate student models distilled from the Pythia and Qwen2.5 families on two QA benchmarks, SQuAD and MLQA, under zero-shot and one-shot prompting conditions. Results show that student models retain over 90% of their teacher models' performance while reducing parameter counts by up to 57.1%. Furthermore, one-shot prompting yields additional performance gains over zero-shot setups for both model families. These findings underscore the trade-off between model efficiency and task performance, demonstrating that KD, combined with minimal prompting, can yield compact yet capable QA systems suitable for resource-constrained applications.

Keywords: *LLM; Knowledge Distillation*

11.1 Introduction

In recent years, the increasing capabilities of Large Language Models (LLMs) have significantly advanced the field of NLP, enabling state-of-the-art results in tasks ranging from Question Answering (QA) [58] to summarization [294] and translation [209]. However, their considerable memory and compute demands hinder practical deployment, particularly in resource-constrained environments [160] [180]. These challenges have fueled interest in model compression techniques, which aim to reduce model size while preserving the performance of large models.

Among these methods, Knowledge Distillation (KD) [154] has emerged as a widely adopted strategy for transferring the behavior of a larger teacher model to a smaller student model. While prior studies have demonstrated its effectiveness across various NLP tasks, it remains unclear to what extent the compression is possible without degrading task-specific performance.

This work studies the trade-offs between model size and task performance by distilling models of varying sizes from the *Pythia* [37] and *Qwen2.5* [375] families. To complement this scaling analysis, we evaluate how lightweight

prompting strategies —i.e., zero-shot and one-shot prompting —impact model behaviour. Our contributions are as follows: (1) We perform a scaling analysis by distilling multiple student models from larger teacher models and evaluating their performance on QA tasks, (2) We assess these models under zero-shot and one-shot prompting conditions, offering insights into the role of lightweight prompting in compressed models.

11.2 Related Work

KD has proven effective across a variety of NLP tasks, including classification and sequence generation, with notable examples such as DistilBERT [305] and TinyBERT [176]. For autoregressive models, the DISTILLM framework [190] proposed Skew Kullback-Leibler Divergence and off-policy sampling strategies to improve stability and distillation quality.

Our work builds directly on DISTILLM by applying its distillation methodology to the Pythia [37] and Qwen2.5 [375] model families.

In parallel, few-shot prompting, a form of In-Context Learning (ICL), allows models to generalize to new tasks by conditioning on a small number of examples in the input prompt [58]. Recent work has begun to explore the intersection of KD and ICL. Notably, **(author?)** [164] introduced in-context learning distillation to transfer few-shot capabilities from larger to smaller models, focusing on meta-trained models evaluated on multitask and instruction-tuned benchmarks, CrossFit [379] and LAMA [281]. Similarly, **(author?)** [307] examined KD for few-shot intent classification.

Our study differs in two key ways: (1) we focus on extractive QA tasks, rather than multitask or classification settings, and (2) we explicitly evaluate performance under zero-shot and one-shot prompting, without relying on instruction tuning or prompt mixture training. This setup allows us to isolate how prompt structure and model size interact, offering new insights into the compression-performance trade-off in QA settings.

11.3 Methodology

We conduct a scaling analysis using distilled models of varying sizes and assess their performance under different prompting strategies.

11.3.1 Data Preprocessing

We conduct our experiments on two widely used QA benchmarks: SQuAD v2.0 [295] and MLQA [210].

SQuAD v2.0 contains over 150K English QA pairs, including a subset of unanswerable questions, which allows us to assess model performance in ambiguous scenarios. MLQA is a multilingual benchmark covering seven languages. In this work, we focus on the English and German subsets to evaluate cross-lingual generalization capabilities.

To study ICL capabilities of the models, we create zero-shot and one-shot variants of each dataset. In the zero-shot setting, each model sees only the input context and question. In the one-shot setting, we prepend a single in-context example to each instance. To avoid overlap, the demonstration set used for constructing one-shot prompts is excluded from both the training and test sets.

Table 49 in the appendix illustrates the dataset formats for both prompting setups.

11.3.2 Prompting Strategy

We used simple few-shot prompting by prepending a single representative example to every instance in the dataset. The prompt format follows a standard QA structure, comprising a context, a question, and an answer, and was kept consistent across all models and datasets to ensure comparability. While more advanced prompting strategies (e.g., chain-of-thought) could yield better performance, we focused on isolating the effect of model compression by applying minimal and uniform prompting across all settings, ensuring a controlled experimental setup.

11.3.3 Model and Dataset Selection

We selected the Pythia [37] and Qwen2.5 [375] model families due to their open-source availability, diverse range of model sizes, and strong baseline performance on QA tasks. During this study, these were among the few publicly available model families with multiple size variants under a unified architecture, making them well-suited for studying model compression trade-offs. Also, both are compatible with the DISTILLM framework, facilitating efficient knowledge distillation and evaluation.

We chose SQuAD v2.0 and MLQA datsets as they represent two widely adopted QA benchmarks with complementary characteristics. SQuAD provides high-quality span-based annotations in English, including a substantial portion of unanswerable questions, which allows for evaluating model robustness under ambiguity. MLQA extends the evaluation to multilingual settings, offering parallel QA examples across multiple languages. We focus on the English and German subsets to assess cross-lingual generalization. Together, these models and datasets form a comprehensive experimental setup for analyzing the effectiveness of knowledge distillation in diverse QA scenarios.

11.3.4 Model Training and Distillation

We distill student models from two teacher models: Qwen2.5-7B and Pythia-2.8B. Each teacher is fine-tuned on the corresponding dataset and prompting configuration (zero-shot and one-shot). KD is then performed using the

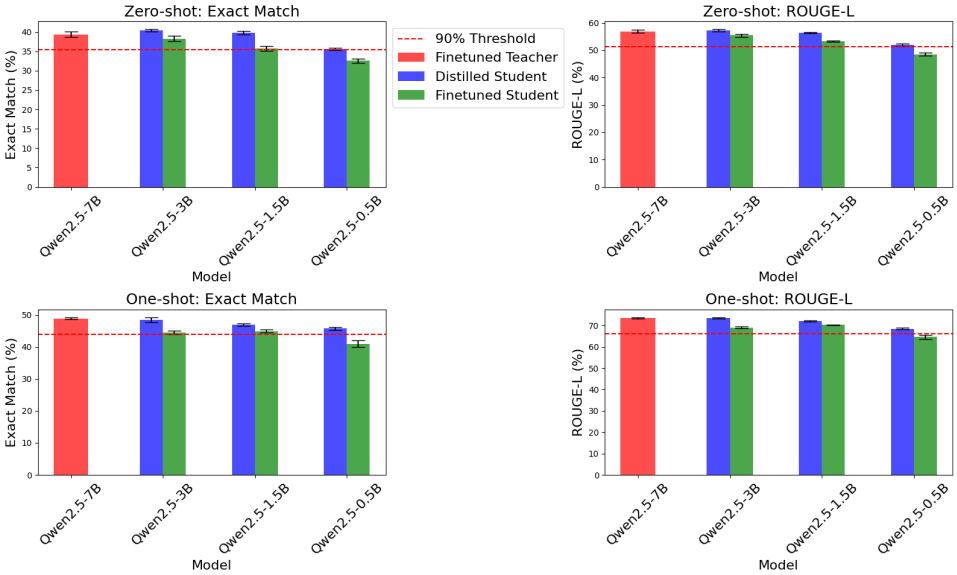


Figure 49: Exact Match and ROUGE-L scores for Qwen2.5 models on the MLQA English split under zero-shot and one-shot settings. Error bars denote standard deviation across five random seeds.

DISTILLM framework [190], transferring the teacher's behavior to smaller student models. For Qwen2.5, student models range from 3B to 0.5B parameters, while for Pythia, they range from 1.4B down to 70M parameters. Full model configurations are provided in the Appendix section (Table 50).

11.3.5 Evaluation Setup

We evaluate all models using Exact Match (EM) [296] and ROUGE-L [120], capturing both answer precision and semantic overlap. To ensure robustness, all experiments are run across five different random seeds, and results are reported as the average across these runs.

We compare: (a) fine-tuned teacher models, (b) distilled student models, and (c) fine-tuned student models.

Specifically, we analyze: (1) how model size impacts performance retention, (2) the effect of prompting strategies on compressed models, (3) the comparative strengths of distillation vs. fine-tuning.

11.4 Results and Discussion

In this section, we present the empirical results from our experiments, analyzing the effects of model compression and prompting strategies on QA performance.

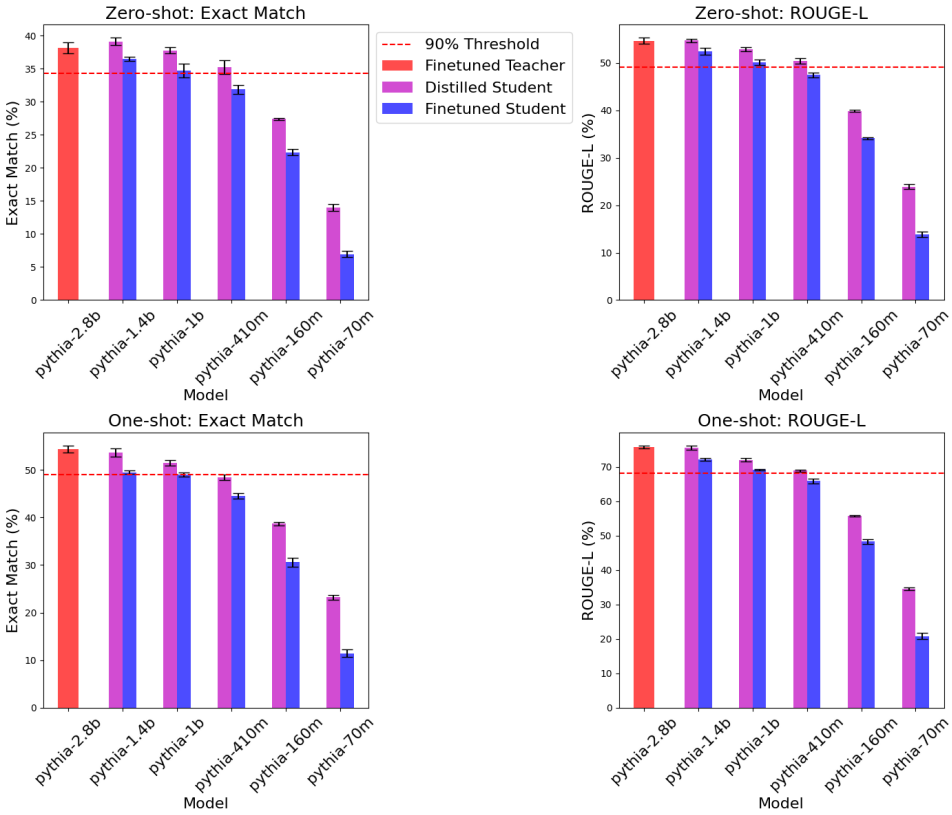


Figure 50: Exact Match and ROUGE-L scores for Pythia models on the MLQA English split under zero-shot and one-shot settings. Error bars denote standard deviation across five random seeds.

Model Size vs. Performance: Figures 49 and 50 illustrate how model performance scales with size across the Qwen2.5 and Pythia model families under both zero-shot and one-shot prompting conditions. We observe that larger student models, such as Qwen2.5-3B and Pythia-1.4B, consistently retain over 90% of their teacher models' performance in both EM and ROUGE-L metrics. In particular, the distilled Qwen2.5-3B closely matches or even slightly exceeds the performance of the Qwen2.5-7B teacher in zero-shot settings, suggesting that moderate compression can enhance generalization.

As the model size decreases, performance degrades more noticeably, especially for smaller models like Qwen2.5-0.5B and Pythia-70M. This trend is consistent across both zero-shot and one-shot settings, suggesting that maintaining strong task-specific capabilities under compression becomes increasingly challenging at lower parameter scales. Nevertheless, moderately sized student models keep a good balance between efficiency and performance, making them good candidates for resource-constrained applications.

All experiments were run with five different random seeds, with results

showing low variance and high reproducibility across runs (see Appendix B.3 for detailed variance analysis).

Distillation vs. Fine-Tuning: *Distilled models generally outperform their fine-tuned counterparts.* For instance, the distilled Qwen2.5-3B and Pythia-1.4B models achieve substantially higher scores compared to their fine-tuned counterparts, narrowing the gap to their teacher models. In contrast, fine-tuned students show larger performance degradation, particularly at smaller scales (e.g., Qwen2.5-0.5B and Pythia-70M). These results demonstrate that KD not only compresses models effectively but also enhances their ability to generalize from prompts, making it a more reliable strategy for scaling down LLMs without significant task-specific performance loss.

Effect of Prompting (Zero-shot vs. One-shot): We observe that one-shot prompting generally improves performance across most datasets and models. As shown in Table 47, student models like Qwen2.5-3B and Pythia-1.4B gain substantial EM and ROUGE-L improvements under one-shot prompting compared to zero-shot settings.

For instance, Qwen2.5-3B achieves a +5.4 EM point improvement on the MLQA German split, and Pythia-1.4B gains +14.5 EM points on the MLQA English split. These findings suggest that providing an in-context example can significantly enhance the generalization ability of compressed models.

Another key observation is the variability in performance retention across languages: Qwen2.5-3B performs robustly even on the MLQA German subset, showing stronger multilingual generalization than the Pythia models, which were evaluated only on English. These results highlight not only the feasibility of compression but also the importance of model family and language in shaping student model effectiveness.

Prompt Sensitivity and Evaluation Discrepancies: The impact of prompting is not uniformly positive across all models. Notably, we observed an inconsistency with Pythia models on the SQuAD dataset, where one-shot prompting improved performance during validation but led to significant performance degradation during evaluation. As shown in Table 48, the Pythia-1.4B model achieved substantially better performance with one-shot prompting during validation; however, on the test set, the same model significantly underperformed in the one-shot setting compared to zero-shot.

This discrepancy suggests that while the model could effectively leverage in-context examples during training and validation, it failed to generalize that behavior to the test set. Several factors may contribute to this phenomenon:

- **Prompt length sensitivity**: The longer prompts in one-shot settings may affect model behavior differently depending on the specifics of the

Dataset	Setting	Qwen2.5 Models						Pythia Models					
		Teacher (7B)		Best Student				Teacher (2.8B)		Best Student			
		EM	R-L	EM	R-L	Model	Size	EM	R-L	EM	R-L	Model	Size
MLQA (EN)	Zero	39.32	56.92	**40.33**	**57.28**	3B	43%	38.14	54.66	**39.15**	**54.76**	1.4B	50%
	One	48.87	73.34	48.45	73.47	3B	43%	54.36	75.72	53.65	75.53	1.4B	50%
MLQA (DE)	Zero	26.25	43.19	**28.32**	**44.38**	3B	43%	-	-	-	-	-	-
	One	29.17	55.21	**32.91**	**57.87**	3B	43%	-	-	-	-	-	-
SQuAD	Zero	62.51	73.29	**64.09**	**74.50**	3B	43%	59.78	70.46	**59.86**	69.72	1.4B	50%
	One	65.03	78.37	**65.23**	78.05	3B	43%	43.23	66.36	42.86	66.10	410M	15%

Table 47: Top-performing student models retain $\geq 90\%$ of their teacher's performance under Exact Match (EM) and ROUGE-L (R-L) metrics. Bold values indicate where students outperform teachers. The size column shows the student model size as a percentage of the teacher model. The dash (-) indicates no experiments were conducted.

Split	Setting	EM	ROUGE-L
Validation	Zero-shot	63.60	73.18
	One-shot	**68.00**	**78.74**
Test	Zero-shot	**59.86**	**69.72**
	One-shot	43.23	66.36

Table 48: Validation vs. evaluation scores of Pythia-1.4B on the SQuAD dataset. Despite a strong one-shot performance during validation, the evaluation score drops significantly in the one-shot setting.

validation vs. test examples.

- **Overfitting to few-shot structure**: The model may have overfitted to the specific structure or patterns in the validation examples.
- **Distribution shifts**: Subtle differences in example distributions between validation and test splits may affect how well in-context learning generalizes.

This finding highlights the importance of comprehensive evaluation setups when assessing few-shot learning capabilities and suggests that prompt engineering and robust validation procedures are crucial when deploying knowledge-distilled models in real-world scenarios.

Implications and Observations: This study yields several important observations about the relationship between model size, prompting strategies, and the effectiveness of KD in QA tasks.

First, our results demonstrate that KD enables student models to retain strong ICL capabilities. For example, Qwen2.5-3B and Pythia-1.4B consistently achieved over 90% of their teacher models' performance with significantly fewer parameters. This suggests that distillation is a practical strategy for compressing LLMs without substantially sacrificing performance, making these models more suitable for deployment in resource-constrained environments.

Second, we observe that the benefit of prompting strategies is not uniform across model sizes. Larger models, such as Qwen2.5-3B, benefit consistently from one-shot prompting, particularly on cross-lingual QA tasks like MLQA (German). In contrast, smaller models, such as Pythia-70M, struggle to utilize in-context examples effectively, even when overall variance is low. This suggests a capacity threshold, below which few-shot learning becomes less effective, highlighting the importance of considering model size when applying ICL.

Finally, we observe inconsistencies between validation and test performance in some cases, most notably for Pythia models on the SQuAD dataset.

While one-shot prompting improved validation results, the same setup led to underperformance during evaluation. This points to a potential sensitivity to prompt structure or dataset distribution differences across splits. It highlights the need for careful prompt design and evaluation setup when assessing few-shot learning behavior.

Taken together, these findings suggest that while distillation offers a promising path toward model efficiency, the interaction between compression and ICL behavior needs further investigation, particularly in settings involving diverse prompting strategies such as instruction-based or chain-of-thought formats.

Limitations

While this study offers valuable insights into compressing LLMs through KD under zero-shot and one-shot prompting, it also shows limitations that suggest avenues for future research. We focus only on simple prompting strategies based on single example concatenation, without exploring more advanced formats such as chain-of-thought or instruction-based prompting, which may further enhance ICL capabilities.

Additionally, we observe inconsistencies between validation and test performance, particularly for Pythia models on SQuAD, suggesting sensitivity to prompt structure and dataset splits. This points to potential issues with generalization and prompt sensitivity, which need further investigation—especially with respect to prompt design, dataset splits, and overfitting to seen examples.

Finally, although this work centers on compression via KD, combining it with other techniques such as quantization or parameter-efficient fine-tuning (e.g., LoRA, adapters) could further improve model deployability and efficiency.

11.5 Conclusion

This study offers a detailed investigation into compressing LLMs through KD, with a focus on retaining their ICL abilities. Evaluating models from the Pythia and Qwen2.5 families on both SQuAD and MLQA datasets, we find that distilled student models can retain over 90% of their teacher models' performance while achieving substantial reductions in parameter size. One-shot prompting further increases performance, particularly in multilingual settings like MLQA (German), highlighting the usefulness of few-shot learning in compressed models. Overall, our findings suggest that KD, when combined with few-shot prompting, offers a promising direction for building compact, generalizable, and cost-efficient language models suitable for real-world deployment.

While our study focuses on extractive QA tasks, the observed trends,

particularly the strong performance retention of distilled models under minimal few-shot prompting, may generalize to other NLP tasks that can be framed as QA problems. Recent work on instruction-tuned models such as UnifiedQA [185] and FLAN [356] has shown that a wide range of tasks, including sentiment analysis, text classification, and natural language inference, can be effectively reformulated using a QA-style input-output format. Moreover, both the Pythia and Qwen2.5 model families have demonstrated strong baseline performance on non-QA benchmarks, such as LAMBADA, PIQA, and WinoGrande for Pythia [37], and MMLU, BBH, and ARC-C for Qwen2.5 [375], highlighting their versatility beyond QA. Therefore, our findings may hold relevance for broader instruction-style applications, especially under few-shot or zero-shot settings. Future work could explore whether similar compression and prompting strategies yield consistent gains across diverse task families.

Appendix B:

B.1 Dataset Examples

We provide additional details on the format of zero-shot and one-shot prompts used in our experiments.

B.2 Model Configurations and Architectures

In our scaling analysis, we experimented with various model sizes to investigate the trade-off between parameter count and performance. For each teacher model, we created several student models with progressively decreasing parameter counts. Table 50 provides the complete details of all teacher and student models used in our experiments. The Qwen2.5 student models range from 3B parameters (43% of teacher size) down to 0.5B parameters (7% of teacher size), while the Pythia student models range from 1.4B parameters (50% of teacher size) down to 70M parameters (2.5% of teacher size).

B.3 Variance Analysis

Table 51 reports the mean and standard deviation of EM and ROUGE-L across five runs with different random seeds for distilled student models from both model families. We find consistent results with low variance, which indicates that our findings are stable and reproducible across different training runs.

Notably, the smallest model in the Pythia family, Pythia-70M, shows low variance across runs, indicating stable behavior. Although its overall performance is lower than larger models, its consistency suggests that performance

Zero-shot Dataset	
id	d8b0801e5f6428a965bec3977ee2f0d86e6146a0
prompt	**Context:** 2001: Classic Bruce Willis: The Universal Masters Collection (Polygram Int'l, OCLC 71124889) **Question:** In what year was the Classic Bruce Willis collection released?
output	2001
One-shot Dataset	
id	d8b0801e5f6428a965bec3977ee2f0d86e6146a0
prompt	**Example:** **Context:** In March 1010 his successor, Basil Mesardonites, disembarked with reinforcements and besieged the rebels in the city... **Question:** When did Mesardonites leave? **Answer:** March 1010 **Current Task:** **Context:** 2001: Classic Bruce Willis: The Universal Masters Collection (Polygram Int'l, OCLC 71124889) **Question:** In what year was the Classic Bruce Willis collection released?
output	2001

Table 49: Preprocessed zero-shot and one-shot examples.

Teacher Model	Student Models
Qwen2.5-7B	Qwen2.5-3B Qwen2.5-1.5B Qwen2.5-0.5B
Pythia-2.8B	Pythia-1.4B Pythia-1B Pythia-410M Pythia-160M Pythia-70M

Table 50: Teacher and student model configurations evaluated in our experiments.

stability does not necessarily correlate with high accuracy. This highlights that even low-capacity models can produce reliable outputs, further supporting the robustness of our evaluation setup.

Model	Dataset	EM (±std)	ROUGE-L (±std)
Qwen2.5-3B	MLQA-EN (Zero)	40.33 ± 0.31	57.28 ± 0.36
Qwen2.5-3B	MLQA-EN (One)	48.45 ± 0.76	73.47 ± 0.30
Qwen2.5-1.5B	MLQA-EN (Zero)	39.74 ± 0.49	56.45 ± 0.20
Qwen2.5-1.5B	MLQA-EN (One)	46.97 ± 0.37	72.04 ± 0.41
Pythia-1.4B	MLQA-EN (Zero)	39.15 ± 0.54	54.76 ± 0.35
Pythia-1.4B	MLQA-EN (One)	53.65 ± 0.85	75.53 ± 0.63
Pythia-70M	MLQA-EN (Zero)	13.97 ± 0.53	23.90 ± 0.53
Pythia-70M	MLQA-EN (One)	23.21 ± 0.52	34.64 ± 0.39

Table 51: Variance analysis (mean ± std) for student models across five seeds.

12 Paper 13: LLM Fine Tuning

ELMTEX: Fine-Tuning Large Language Models for Structured Clinical Information Extraction. A Case Study on Clinical Reports

*Aynur Guluzade, Naguib Heiba, **Zeyd Boukhers**, Florim Hamiti, Jahid Hasan Polash, Yehya Mohamad, and Carlos A. Velasco*

(DOI: 10.1007/978-3-031-95841-0_34)

Abstract Europe's healthcare systems require enhanced interoperability and digitalization, driving a demand for innovative solutions to process legacy clinical data. This paper presents the results of our project, which aims to leverage Large Language Models (LLMs) to extract structured information from unstructured clinical reports, focusing on patient history, diagnoses, treatments, and other predefined categories. We developed a workflow with a user interface and evaluated LLMs of varying sizes through prompting strategies and fine-tuning. Our results show that fine-tuned smaller models match or surpass larger counterparts in performance, offering efficiency for resource-limited settings. A new dataset of 60,000 annotated English clinical summaries and 24,000 German translations was validated with automated and manual checks. The evaluations used ROUGE, BERTScore, and entity-level metrics. The work highlights the approach's viability and outlines future improvements.

Keywords: *LLM; Fine-Tuning, Information Extraction, Digital Health Applicaiton*

12.1 Introduction

There is a growing need for interoperability and digitalization in Europe [104]. The variety of health systems and the need to cope with legacy documentation and procedures present in the sector foster the search for innovative approaches, which should address the upcoming requirements of the European Health Data Space (EHDS [107, 171]). Recent advances in Large Language Models (LLMs) offer a potential for structured clinical information extraction (IE) to enhance the quality and interoperability of healthcare. Furthermore, automated IE reduces the manual effort required by healthcare professionals in daily tasks, such as clinical decision-making, research, and operational procedures, which require a rapid extraction of structured and standardized information, addressing the specialized language and terminologies used in medical documents [205].

The work presented in this paper focuses on the optimization of LLMs to extract structured and standardized information from text-based clinical reports. This work is part of an internal wider project of our Institute in collaboration with clinical teams called ELMTEX, which included a workflow allowing the processing of legacy documentation, sometimes available as images of text documents, which were scanned and automatically transformed into text information, and the development of a user interface that allows medical teams to validate the extracted data. These additional components are not addressed in this paper.

We optimized the models to extract information on multiple categories. We evaluated open-source LLMs of various sizes, first without fine-tuning by only prompting them, and subsequently fine-tuning smaller ones (Llama 3.1 8B, Llama 3.2 1B & 3B [130]) to observe their performance improvement. We assessed the impact of fine-tuning on the accuracy of IE from clinical reports using different metrics, which include a wide range of evaluation metrics, from n-gram-based to similarity-based and entity-level metrics. We explored methods ranging from basic prompting to advanced prompting with in-context learning, as well as LoRA fine-tuning. Experiments indicated that fine-tuning Small Language Models (SLMs) achieves better results compared to existing large models.

To support these experiments, we introduced a new large corpus comprising $60,000$ annotated clinical reports in English and $24,000$ in German. Each instance includes a summary extracted from PubMed Central articles[56] and its corresponding structured information in JSON format. The dataset spans diverse categories, including patient family/social history, medical history, diagnosis, and outcome assessment, ensuring comprehensive coverage of key medical information. To facilitate reproducibility and reuse, our code implementation[57] and the dataset [136] are publicly available.

[56] `https://pmc.ncbi.nlm.nih.gov/`

[57] `https://gitlab.cc-asp.fraunhofer.de/health-open/elmtex`

12.2 State-of-the-Art

Earlier approaches for data extraction from clinical reports include rule-based systems [250]. New approaches involve supervised machine learning (ML) models [8, 139]. The constant evolution in the medical field requires the update of such methods, implying a manual effort [9]. Likewise, supervised ML models also need large annotated datasets, which are expensive and take a long time to create [383].

The rising development of Large Language Models (LLMs) in recent years has shown significant potential in Natural Language Processing (NLP), including clinical IE. Transformer-based LLMs [350] can handle and understand large amounts of text with limited task-specific training without the need for a large labeled dataset [291]. A concern is that these models are pre-trained on general texts and they often lack domain-specific knowledge. This can cause LLMs to hallucinate and generate plausible but inaccurate information because the model faces difficulties in understanding medical reports terminology [174].

Clinical text has a different syntax and vocabulary compared to general text [365], leading to the development of domain-specific models. Early efforts included clinical word embeddings [367], and later models such as ClinicalBERT, SciBERT, BioBERT, and PubMedBERT were inspired by BERT [95]. However, some studies report only minimal performance gains over classical methods such as random forest [72], and LLMs like GPT-3 struggle to achieve competitive results in biomedical NLP tasks [244]. Recent studies have shown that LLMs such as GPT-3.5 and GPT-4 exhibit strong performance in zero- and few-shot learning scenarios for clinical IE tasks, even without domain-specific training [162].

However, concerns persist about their accuracy and potential to generate plausible yet incorrect information (hallucinations [267]). To address these challenges, researchers have explored fine-tuning LLMs on domain-specific data. For example, Gema *et al.* [122] introduces a two-step parameter-efficient fine-tuning framework for LLMs in clinical applications; Wang *et al.* [353] presents a language model optimized for clinical scenarios through fine-tuning with real-world medical data; and Peng *et al.* [274] evaluate soft prompt-based learning algorithms for LLMs in clinical concepts and relation extraction.

All these approaches rely on the availability of high-quality datasets which is crucial for advancing the task. Open access electronic health record (EHR) datasets, such as those provided by the MIMIC database, have been instrumental in developing and evaluating NLP models [263]. However, the scarcity of large-scale annotated clinical datasets, particularly for non-English languages, remains a significant barrier. Efforts are underway to create synthetic datasets and translate existing ones to mitigate this [301].

12.3 ELMTEX Dataset and Evaluation Approach

For a clinical report R, represented as a sequence of tokens $R = \{r_1, r_2, \ldots, r_n\}$, the task is to map R to a structured representation S:

$$S = \{C_1 : s_1, C_2 : s_2, \ldots, C_k : s_k\} \tag{59}$$

where S is an object containing predefined categories; $C = \{C_1, C_2, \ldots, C_k\}$ represents the set of categories with $k = 15$; and s_i is the extracted information for category C_i, expressed as a string. For each s_i, we use separate concepts delimited by semicolons (;) since each category may contain multiple concepts. The categories C_i are listed in Table 52.

The goal is to train and evaluate LLM f_θ, to approximate the mapping function $f_\theta : R \to S$, such that for each category C_i, the extracted string s_i accurately reflects the corresponding structured information derived from R. The dataset $\mathcal{D}$ used for training and evaluation consists of N samples, where each sample (R, S^*) includes a clinical report R and its corresponding ground truth structured representation S^*.

12.3.1 Evaluation

The experiments explore three model setups, described in the following:

Naive Prompting. It consists of directly querying the LLM with a simple instruction to extract information for all categories from R. We constructed a simple prompt P to describe the task and explicitly list the categories C from which information needs to be extracted, without providing detailed definitions or scope for each category. The expectation is that the LLM f_θ can infer the meaning and scope of each category based solely on the category names and perform the task as instructed.

This setup relies on the LLM's inherent ability to understand the semantic meaning of the task described in the prompt, comprehend the intent behind each category name, and generate syntactically valid and semantically accurate structured output without further guidance. However, due to the lack of explicit definitions or examples for each category, the model's output may vary significantly in quality and completeness. The performance in this setup depends primarily on the model's pre-training data and generalization capabilities.

Advanced Prompting with In-Context Learning. It includes examples of input-output pairs as context within the prompt to guide the LLM. Here, we explicitly defined each category $C_i \in C$, providing clear descriptions and the scope of information expected for each category. To further improve task performance, we integrated in-context learning by incorporating examples retrieved from a training set. Given a clinical report R from the test set,

an encoder-based retrieval model g_ϕ retrieves the top m most similar clinical reports $\mathcal{R}' = \{R_1, R_2, \ldots, R_m\}$ from the training set, where the training set is disjoint from the test set. These retrieved reports are paired with their corresponding annotated structured representations $\mathcal{S}' = \{S_1^*, S_2^*, \ldots, S_m^*\}$, forming the in-context examples. Formally, the retrieval process is defined as:

$$g_\phi : R \to \{(R_1, S_1^*), (R_2, S_2^*), \ldots, (R_m, S_m^*)\} \tag{60}$$

where g_ϕ ranks training examples based on their semantic similarity to R using cosine similarity. The prompt P is constructed with a detailed task description, instructing the LLM to extract structured information from R. The explicit definitions and scopes for all categories C ensure clarity in the expected output for each category. In addition, the retrieved clinical reports $\mathcal{R}'$ and their corresponding annotated structured representations $\mathcal{S}'$ are appended to the prompt as in-context examples, demonstrating how the task should be performed. Our objective is to leverage the LLM to use explicit task instructions and category definitions to better understand the task. We intended that the LLM learns from the retrieved in-context examples to handle the clinical report of the test R more effectively and to generalize to new instances using similar previous examples. By integrating retrieval-augmented in-context learning, this setup mitigates ambiguities in naive prompting and enables the LLM to produce a more accurate and reliable output for each category.

LLM Fine-tuning. We trained small LLMs on a domain-specific dataset to optimize f_θ for clinical reports. Building upon our naive prompting approach, we employed parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA) [161]. This method enables the LLM f_θ, to adapt to the task while maintaining its general pre-training capabilities. The goal is for the LLM to learn the task-specific mapping $f_\theta : R \to S$ and the definitions and scope of each category $C_i \in C$ during fine-tuning, therefore eliminating the need for detailed prompts at inference time. Fine-tuning is conducted on the training set, which comprises 90% of the total dataset $\mathcal{D}$. Each training instance consists of a clinical report R and its corresponding ground truth structured representation S^*. The objective of fine-tuning is to minimize the loss $\mathcal{L}$ between the predicted structured representation S and the ground truth S^*:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \text{Loss}(f_\theta(R_i), S_i^*) \tag{61}$$

where N is the number of training samples, and Loss refers to the loss of cross-entropy. The fine-tuning process enables the LLM to internalize the

task-specific knowledge required, learn the definitions and scopes of each category C_i directly from the training data, and generalize effectively to unseen test samples, including edge cases and ambiguous scenarios.

By fine-tuning on domain-specific data, the model becomes more robust and precise, overcoming the limitations of relying solely on prompt engineering. This is particularly important given the complexity and variability of clinical reports, where complete coverage of all possible scenarios through prompts alone is unfeasible. Unlike the other setups, which depend heavily on prompt construction, the fine-tuned LLM requires only minimal instructions at inference time.

12.3.2 Dataset Generation Workflow

We introduced a new dataset of clinical report summaries, annotated with structured information across 15 categories [136]. This dataset was created to address the lack of large-scale resources for clinical IE. It also promotes the development of methods tailored to clinical data, helping to improve healthcare provision. The dataset contains $60,000$ annotated English clinical report summaries, from which we translated over $24,000$ examples into German. The dataset is based on PMC-Patients [394], a collection of $167,000$ patient summaries from case reports in PubMed Central. We extracted a subset of patient reports for our work and used a semi-automated approach to create the dataset. First, we defined the categories and their scopes by reviewing related work [63] and consulting physicians to ensure that the selected categories were relevant. Afterwards, we manually annotated reports to serve as in-context learning examples. We then used the GPT-4 model with advanced prompting and in-context learning to generate the initial annotations.

Table 52 provides the dataset statistics for each category, including error rates, derived from samples used for manual validation. Note that not all 15 categories are present in every patient report; their presence depends on the details of the report. To reflect this, we also include the percentage of reports in which each category is present. Please refer to our Appendix [135] for examples of the prompts used, and the results on the German dataset.

12.4 Experiments

12.4.1 Experimental setup

Baseline Models We used the Llama 3 series [130] of models for our experiments. For small-sized models, we employed Llama 3.2 1B&3B Instruct for advanced prompting with in-context learning and LoRA fine-tuning setups. We skipped naive prompting for these models, as their size and pre-training were insufficient for effectively performing the task or generating properly formatted JSON outputs. For medium-sized models, we used Llama 3.1 8B

Table 52: Annotated categories with their presence percentages and corresponding error rates in the English dataset.

Category	Presence (%)	Error Rate (%)
C_1: *age*	100	0.00
C_2: *comorbidities*	37.47	11.00
C_3: *diagnosis*	98.63	8.00
C_4: *diagnostic_procedures*	98.87	1.67
C_5: *family_history*	17.86	0.67
C_6: *gender*	100	0.00
C_7: *interventional_therapy*	73.30	4.17
C_8: *laboratory_values*	67.75	2.67
C_9: *life_style*	22.70	7.00
C_{10}: *medical_surgical_history*	84.29	7.50
C_{11}: *pathology*	73.56	4.67
C_{12}: *patient_outcome_assessment*	92.88	1.00
C_{13}: *pharmacological_therapy*	70.33	1.50
C_{14}: *signs_symptoms*	95.96	2.33
C_{15}: *social_history*	7.94	7.00

Instruct across all setups, including naive prompting, advanced prompting, and fine-tuning. For large models, we selected Llama 3.1 70B Instruct and Llama 3.1 405B Instruct. These were tested only with naive and advanced prompting, as fine-tuning such large models is not optimal for a task of this specificity. For the Llama 405B Instruct model, we used FP8 quantization due to GPU resource limitations. The experiments were primarily run on 4 H100 GPUs. However, smaller and medium-sized models could be run on a single GPU, as LoRA fine-tuning reduces the GPU memory requirements.

Evaluations Metrics We used three metrics for evaluation, chosen to provide complementary insights into the accuracy and relevance of the extracted information. We evaluated each category independently, averaging the results across all categories at the end. First, we used ROUGE to measure the n-gram overlap (unigrams, bigrams, and longest common subsequence) between the model's output and reference summaries, assessing content and structural alignment. Second, BERTScore was used to evaluate the semantic similarity using contextual embeddings. We calculated precision, recall, and F1 scores based on cosine similarity, to ensure that the generated summaries captured the intended meaning. Finally, we considered the Entity-level metrics, focused on clinical accuracy by extracting entities such as medications and diagnoses with a medical NER model[58]. We calculated precision, recall,

[58] `https://github.com/allenai/scispacy`

and F1 scores to compare the extracted entities between the outputs and references.

Each of these metrics was chosen to target specific aspects of the evaluation: ROUGE for surface-level alignment, BERTScore for semantic understanding, and entity-level metrics for domain-specific accuracy. This combination provided a well-rounded assessment of the performance of the model.

12.4.2 Results

Table 53 summarizes the results comparing different setups in the models. We observe that the LLama 3.1 8B fine-tuned model achieves the best overall performance across all metrics, outperforming all non-fine-tuned models, including the LLama 405B, even with advanced prompting. This suggests that fine-tuning is crucial for enabling the model to fully grasp the various definitions and concepts required for each category. We also see that fine-tuning smaller models like LLama 3.2 1B&3B yields surprisingly strong results. These models, which can run on edge devices and require far fewer resources, demonstrate their potential for practical deployment in resource-constrained environments.

On the other hand, we note that only large models perform well with naive prompting, which confirms their inherent advantage due to their scale and pre-training. Advanced prompting and in-context learning, however, enable medium and large models to perform significantly better, underscoring the value of this more sophisticated approach.

12.4.3 Error Analysis

For the error analysis, we randomly sampled 20 incorrect predictions for each model experiment setup and examined the types of errors. The analysis revealed two main error types:

- **Missing Extracted Information** Models often missed specific concepts within certain categories. This was particularly common in naive and advanced prompting setups. While the main concepts for each category were usually extracted, additional relevant concepts were often missed. This error was mainly observed in categories like diagnostic_techniques_procedures, diagnosis, laboratory_values, and pharmacological_therapy. A possible reason is that these categories can involve lengthy lists of concepts, which can be spread over multiple sentences in clinical reports, making it challenging for the LLM to capture all relevant details.

- **Wrongly Categorized Concepts** Errors involving the misplacement or incorrect categorization of concepts were observed, particularly between similar categories. This issue was more frequent in naive and ad-

Table 53: Model performance comparison on our English dataset.

	Models	ROUGE			BERTSc.	Entity-Level		
		R-1	R-2	R-L	F1	P	R	F1
Naive	Llama-3.1-8B-Instr.	0.5585	0.4303	0.5432	0.5949	0.6592	0.5548	0.6025
	Llama-3.1-70B-Instr.	0.5837	0.4772	0.5989	0.6359	0.6773	0.5833	0.6267
	Llama-3.1-405B-Instr.	0.6183	0.5137	0.6261	0.6696	0.7012	0.6025	0.6481
Adv.+ICL	Llama-3.2-1B-Instr.	0.2333	0.1558	0.2255	0.3396	0.4298	0.2564	0.3211
	Llama-3.2-3B-Instr.	0.4112	0.3105	0.3917	0.5553	0.5879	0.4861	0.5321
	Llama-3.1-8B-Instr.	0.6244	0.5178	0.6033	0.6986	0.6818	0.6872	0.6844
	Llama-3.1-70B-Instr.	0.6993	0.5865	0.6792	0.7379	0.7099	0.7392	0.7242
	Llama-3.1-405B-Instr.	0.6969	0.5716	0.6714	0.7287	0.7312	0.7407	0.7359
Fine-t.	Llama-3.2-1B-Instr.	0.7416	0.6350	0.7247	0.7857	0.7595	0.7502	0.7548
	Llama-3.2-3B-Instr.	0.7721	0.6727	0.7555	0.8122	0.7853	**0.7840**	0.7846
	Llama-3.1-8B-Instr.	**0.7771**	**0.6841**	**0.7626**	**0.8253**	**0.7917**	0.7822	**0.7869**

vanced prompting setups and was the primary error type for fine-tuned smaller LLMs (1B&3B). The affected categories included life_style, family_history, social_history, medical_surgical_history, signs_symptoms, and comorbidities. Such errors can occur due to misinterpretation of the LLMs in sentences.
Additionally, for LLama 3.2 1B with advanced prompting, we identified instances of complete hallucination, where the model generated non-existent concepts or concepts copied from in-context learning examples. This behavior was expected given the small size of the model and its limited capacity to retain pre-trained knowledge and follow detailed instructions.

12.5 Conclusions and future work

In this article, we investigated the potential of LLMs for clinical IE, focusing on tasks involving structured data generation from unstructured clinical reports. We systematically evaluated approaches like naive prompting, advanced prompting with in-context learning, and fine-tuning across LLMs of varying sizes, and observed that fine-tuning not only enhances performance but also bridges the gap between large and small LLMs. The results highlight the practical effectiveness of smaller fine-tuned models for deployment in real-world clinical settings with limited resources. To support our experiments, we released a large-scale clinical dataset containing $60,000$ patient summary reports in English. The dataset offers extensive coverage of various clinical categories and serves as a valuable benchmark in the field.

Our future work is focused on different aspects. First, we are refining our demonstrator user interface to facilitate clinicians the evaluation of the results of the model. We also identified that the categories need to be refined to be better synchronized with the standard terminologies and data models in the health sector, as highlighted by some of the approached clinical teams. This will contribute to the refinement of the domain-specific training. Additionally, we are considering the expansion of the multilingual capabilities of the dataset and the training approach. Furthermore, we should also consider how our developments are influenced by the AI-Act [106] in Europe.

13 Paper 14: Topic Modeling

Analyzing the Influence of Hyper-parameters and Regularizers of Topic Modeling in Terms of Renyi Entropy

Sergei Koltcov, Vera Ignatenko, ***Zeyd Boukhers****, and Steffen Staab*

(DOI: 10.3390/e22040394)

Abstract Topic modeling is a popular technique for clustering large collections of text documents. A variety of different types of regularization is implemented in topic modeling. In this paper, we propose a novel approach for analyzing the influence of different regularization types on results of topic modeling. Based on Renyi entropy, this approach is inspired by the concepts from statistical physics, where an inferred topical structure of a collection can be considered an information statistical system residing in a non-equilibrium state. By testing our approach on four models—Probabilistic Latent Semantic Analysis (pLSA), Additive Regularization of Topic Models (BigARTM), Latent Dirichlet Allocation (LDA) with Gibbs sampling, LDA with variational inference (VLDA)—we, first of all, show that the minimum of Renyi entropy coincides with the "true" number of topics, as determined in two labelled collections. Simultaneously, we find that Hierarchical Dirichlet Process (HDP) model as a well-known approach for topic number optimization fails to detect such optimum. Next, we demonstrate that large values of the regularization coefficient in BigARTM significantly shift the minimum of entropy from the topic number optimum, which effect is not observed for hyper-parameters in LDA with Gibbs sampling. We conclude that regularization may introduce unpredictable distortions into topic models that need further research.

Keywords: *Topic modeling, Renyi entropy, Regularization*

13.1 Introduction

Topic modeling (TM) is one of the recent directions in statistical modeling, which is widely used in different fields such as text analysis [1], mass spectrometry [2], analysis of audio tracks [3], image analysis [4], detection and identification of nuclear isotopes [5] and many other applications. Topic models are based on a number of mathematical techniques which are related to determining hidden distributions in collections of big data. However, procedures which restore hidden distributions, possess a set of parameters such as the number of distributions in a mixture of distributions and regularization parameters. These parameters have to be set explicitly by a user of TM. In addition, the values of regularization parameters affect significantly the results of TM [6]. The problem of determining the optimal values of model parameters is complicated by the following issues. First, values of param-

eters can depend on the content of the analyzed dataset, correspondingly, the values of parameters can be specific for different datasets. Second, the parameters may depend on the size of the dataset. Increasing the size of a dataset makes numerical experiments on determining the optimal values of parameters to be extremely time-consuming. However, increasing the size of datasets leads to the fact that such data become comparable to mesoscopic systems and one can apply models and metrics of statistical physics to such datasets. Moreover, a part of topic models is based on different modifications of Potts model [7].

The task of TM consists of stochastic decomposition of the matrix of occurrences of words in documents (F_{dw}) into two matrices: (1) matrix $\Theta = (\theta_{td})$ containing the distribution of topics by documents; (2) matrix $\Phi = (\phi_{wt})$ containing the distribution of words by topics. However, stochastic matrix decomposition is defined not uniquely but with accuracy up to a non-degenerate transformation [8]. If $F_{dw} = \Phi\Theta$ is a solution then $F_{dw} = (\Phi R)(R^{-1}\Theta)$ is also a solution for all non-degenerate R under which $\Phi^0 = \Phi R$ and $\Theta^0 = R^{-1}\Theta$ are stochastic matrices. In terms of TM, ambiguity in retrieving the solution means that the algorithm starting from different initial approximations will conjugate to different points of the solution set. Namely, if running TM with the same values of parameters on the same dataset, different outputs will be obtained. It is explained by the fact that TM is an ill-posed problem [9]. The general solution to this type of problem is based on adding prior information (regularization) and modifying the sampling procedure. Furthermore, regularization can be achieved by introducing a combination of conjugate functions [1] and different types of regularization procedures [8,10]. TM parameter optimization is a significant problem that still needs an extensive research. As a partial solution, we propose an approach based on the concepts of statistical physics. Here, a collection of documents is considered an information thermodynamic system. For such a system, Renyi entropy can be introduced within the thermodynamic formalism [11] analogously to [7]. We propose an effective and universal (i.e. independent of the type of regularization) concept, based on Renyi entropy [12], for analyzing the influence of regularization on the outcome of TM. Our approach allows us to estimate optimal values of TM hyper-parameters including the number of topics and regularization parameters. Here, the optimal number of topics corresponds to the number of topics determined by encoders who label test datasets. We apply Renyi entropy approach to four topic models and two real datasets, additionally, we consider the output of hierarchical Dirichlet process model (HDP). It is important to note that the proposed approach does not apply to HDP models, which would demand its modification and, therefore, a special research. We compare the results of our approach with a standard metric in the field of machine learning, namely, log-likelihood metric and find that our method is faster and, in addition, allows to estimate the optimal number of topics while log-likelihood does not.

In our work, we do not consider metrics and models related to estimation of interpretability of topic models, e.g., Kullback-Leibler divergence [13], semantic coherence [14], word intrusion [15] and others. Investigation of these metrics deserves a separate paper.

Our paper consists of the following sections. Section 2.2 describes standard metrics of quality which are used for determining parameters of topic models and considers their limitations. Section 2.1 introduces basic notations and assumptions of TM. Section 2.3 describes entropy approach where Renyi entropy is proposed as the criteria to optimize parameters and hyperparameters in topic models. Section 3 presents the experiments carried out on two real datasets. Finally, the overall analysis of the obtained results is presented in Section 4.

13.2 Materials and Methods

13.2.1 Topic Models

Let us briefly discuss some basic ideas behind TM and introduce our notations. TM is based on the following assumptions [16]:

1. Let D be the number of documents in a dataset, W be the number of words.

2. There is a fixed number of topics (T) which are discussed in the dataset.

3. Datasets are regarded as sets of triples (w, d, t) from the space $\widetilde{W} \times \widetilde{D} \times \widetilde{T}$, where $\widetilde{W}$ is the set of words, $\widetilde{D}$ is the set of documents, $\widetilde{T}$ is the set of topics.

4. 'Bag of words'. It is supposed that the order of words in documents and the order of documents in a collection are unimportant for TM.

In TM, word probability in a document $p(w \mid d)$ can be expressed as follows:

$$p(w \mid d) = \sum_{t=1}^{T} \phi_{wt}\theta_{td}, \tag{62}$$

where ϕ_{wt} is the probability of a word w to occur under a topic t, θ_{td} is the probability of a topic t in a document d. Probabilities ϕ_{wt} form a matrix of distribution of words by topics $\Phi = (\phi_{wt})_{w=1,\ldots,W;\, t=1,\ldots,T}$ and probabilities θ_{td} form a matrix of distribution of topics by documents $\Theta = (\theta_{td})_{t=1,\ldots,T;\, d=1,\ldots,D}$. Different types of topic models are related to different regularization algorithms. There are two main approaches in TM, namely: (1) Models which are based on maximum likelihood principle [1], where matrices Φ and Θ are searched by Expectation-Maximization (E-M) algorithm. (2) Models which are related to Monte Carlo methodology (Gibbs sampling)

[17], where ϕ_{wt} and θ_{td} are searched by calculating expectation through Monte-Carlo method. Despite different mathematical approaches of these types of models, both of them produce similar topic solutions [17]. It is notable that topic models, regardless of the inference algorithm, transform the initial homogeneous word-topic distribution to heterogeneous distribution with low entropy. The flat (uniform) distribution is usually used as the initial distribution for LDA version with Gibbs sampling procedure, while random number generator is used for initialization of topic models with EM algorithm. In both cases, the initial distribution provides maximum entropy. During TM, the number of words with high probabilities changes significantly. In general, the output of topic modeling contains a relatively small subset of words with high probabilities (about several percents) while the rest words are assigned with probabilities about zero [18]. It should be noted that, according to numerical experiments, the percentage of highly probable words depends on the magnitude of hyper-parameters of the model and on the number of topics. These observations allows us to build a theoretical approach for analyzing such dependency using concepts of statistical physics. In our numerical experiments, five topic models are considered:

1. Probabilistic Latent Semantic Analysis (pLSA) [19] is a basic model with only one parameter—'number of topics'. Inference method for this model is based on E-M algorithm.

2. Latent Dirichlet Allocation model with Gibbs sampling procedure (LDA GS) [20] can be considered a regularized extension of pLSA, where regularization is based on prior Dirichlet distributions for Θ and Φ with parameters α and β correspondingly. Unlike the above pLSA, the inference in this model is based on Gibbs sampling procedure.

3. Variational Latent Dirichlet Allocation model (VLDA). This model uses variational E-M algorithm [1]. We consider the version of this model where regularization is based only on a prior Dirichlet distribution for Θ with parameter α. Selection of values of α is built in the algorithm.

4. The Additive Regularization of Topic Models (ARTM) [10] with smoothing/sparsing regularizers for matrix Φ (smooth/sparse phi) and matrix Θ (smooth/sparse theta), here termed sparse phi and sparse theta, respectively, is an alternative model to pLSA and LDA. These regularizers allow a user to obtain subsets of topics highly manifest in a small number of texts and/or words (sparsing effect), as well as subsets of topics relatively evenly distributed across all texts and words (smoothing effect). The parameter that controls the value of sparsing is a regularization coefficient termed τ. This model can be considered a regularization of pLSA, where regularization is embedded in E-M algorithm (regularized' E-M algorithm).

5. Hierarchical Dirichlet Process model (HDP) is an alternative approach, providing the possibility to restore hidden topics without selecting the number of topics in advance [21,22]. Although this model is non-parametric, in real scenarios, users need to set some parameters, e.g., truncation on the allowed number of topics in the entire corpus. Since HDP returns the same number of topics as the top-level truncation that is set before, it is assumed that by discarding empty ones, the true number of topics can be obtained [22].

A more detailed description of pLSA, LDA GS, VLDA can be found in [7] (see supplementary material). For description of ARTM, we refer the reader to [10], and for HDP to [21].

13.2.2 Standard Metrics in the Field of Topic Modeling

To estimate the quality of topic models and to determine the values of parameters, three functions are most often employed for this purpose: (1) perplexity, (2) log-likelihood, (3) harmonic mean. The perplexity is a standard metric for estimating the model's predictive capability on new data and can be expressed in the following way [23]:

$$\text{perplexity}(D_{\text{test}}) = \exp\left(-\frac{\sum_{d=1}^{M}\log p(d)}{\sum_{d=1}^{M} N_d}\right) = \exp\left(-\frac{\sum_{d=1}^{M}\sum_{w=1}^{W} n_d^w \log\left(\sum_{t=1}^{T}\phi_{wt}\theta_{td}\right)}{\sum_{d=1}^{M} N_d}\right), \tag{63}$$

where N_d is the number of words in document d, M is the number of test documents, n_d^w is the number of times term w has been observed in document d. The lower the perplexity score is the better the parameters' values are. Perplexity can also be presented as the exponent of Gibbs-Shannon entropy [24,25]. The use of perplexity for the selection of parameters of topic models is discussed in a number of works [1,20,26].

In work [26], the perplexity is used for determining the optimal number of topics. The authors demonstrated that the perplexity decreases monotonously by increasing the number of topics and does not assist in selecting the number of topics. Some works show another behaviour of perplexity, for example, authors of [17] demonstrate that the perplexity as a function of hyper-parameters has a notable unique minimum for LDA GS model, VLDA and LDA with collapsed variational Bayesian inference. Authors of [27] show that the perplexity as a function of the number of topics has a notable minimum for LDA GS model, and maximal values of perplexity correspond to $T \to 1$ and $T \to \infty$. In [28], it has been shown that the perplexity, used for a model with feature regularization, has clear minimum for some values of varying parameters and the maximum of perplexity corresponds to the maximum value of varying parameter. Thus, it can be noticed

that different types of perplexity behaviour can be found in literature on TM without an explanation of such behaviour.

The use of perplexity has some limitations, which are reviewed in [29]. The authors demonstrated that the value of perplexity depends on the vocabulary size of the collection which was used for topic modeling. The dependence of perplexity value on type of topic model and size of vocabulary is shown in [30] as well. Thus, the comparison of topic models by means of perplexity is complicated [27,28] if models were implemented on different datasets and different languages. Therefore perplexity-based methods are not stable.

Another measure, which is often used when analyzing the results of topic modeling, is logarithm of likelihood which can be presented in the following way [23,31]:

$$\ln\left(P(\widetilde{D} \mid \Phi, \Theta)\right) = \sum_{d=1}^{D}\sum_{w=1}^{W} n_d^w \ln\left(\sum_{t=1}^{T} \phi_{wt}\theta_{td}\right), \tag{64}$$

where n_d^w is frequency of word w in document d. Usually, the calculation of this value is carried out when the perplexity stops changing. The hyper-parameters and number of topics are selected when finding maximum of logarithm of likelihood [20]. Notice that logarithm of likelihood is a version of perplexity and different types of log-likelihood behaviour are shown in literature as well as for perplexity.

Harmonic mean is a metric that allows to evaluate how well the model can fit to the data. Considering LDA GS model, harmonic mean can be expressed as follows [32]:

$$HM(\{P(d \mid z^{(s)}, \Phi)\}_{s=1}^{S}) = \left(\frac{1}{S}\sum_{s}\frac{1}{P(d \mid z^{(s)}, \Phi)}\right)^{-1}, \tag{65}$$

where $\{z^{(s)}\}_{s=1}^{S}$ are S samples from a Gibbs sampler after a burn-in period, d is a document. Harmonic mean is used as an estimator of $P(d \mid \Phi, \alpha)$. Despite the fact that harmonic mean method is simple and relatively computationally efficient, authors of many works express doubts about this method [15,32] as an evaluation technique in TM.

Let us mention that there are methods that aim to optimize hyper-parameters in the LDA model [31,33], however, they are based on log-likelihood maximization and do not consider the selection of hyper-parameters values combined with optimizing the number of topics. In addition, such methods were not tested for compliance with human judgements.

13.2.3 Entropy Approach for Analysis of Topic Models

The entropy approach is based on the idea that a large document collection can be considered an information system, for which Renyi entropy can

be calculated in terms of the 'density of states' and internal energy [7]. We theoretically assume and demonstrate experimentally that the optimal number of topics and the optimal values of hyper-parameters correspond to the minimum Renyi entropy. The 'density of states' function can be expressed through the experimentally determined variables in the following way: $\rho = N/(WT)$, where N is the number of words with relatively high probabilities ($p > 1/W$). The internal energy is expressed through the sum of word probabilities in the following way:

$$E = -\ln(\widetilde{P}) = -\ln\left(\sum_{w,t} p(w \mid t) \cdot \mathbf{1}_{\{p(w|t)>1/W\}}\right), \tag{66}$$

where $\mathbf{1}_{\{.\}}$ is an indicator function.

Thus, topic model is described by two observable parameters: (1) the sum of probabilities of highly probable words); (2) the number of highly probable words, N. Therefore, partition function (statistical sum) of a topic model can be expressed as $Z_q = \rho \cdot (q\widetilde{P})^q$, where $q = 1/T$ [34]. Correspondingly, Renyi entropy of a topic model is expressed in terms of partition function as

$$S_q^R = \frac{\ln(Z_q)}{1-q}. \tag{67}$$

A more detailed explanation of formulating Renyi entropy for topic models can be found in [7,34]. Application of Renyi entropy for investigation of TM results is useful due to the following reasons. Firstly, Renyi entropy determines the degree to which the results of TM are non-equilibrium, so it accounts for the contribution of the initial distribution of the topic model. Secondly, topic models can be optimized based on finding the minimum of Renyi entropy. Thirdly, when calculating Renyi entropy, one actually calculates the difference between two processes. Namely, increasing the number of topics, on the one hand, leads to decreasing Gibbs-Shannon entropy and, on the other hand, to increasing internal energy. What follows from this is the existence of an area where these two processes counterbalance each other. In this region, free energy and, correspondingly, Renyi entropy have the minimum values. Minimum of Renyi entropy corresponds to maximum of information of a topic model [7]. Hence, evaluation of the influence of hyper-parameters on the results of TM can be measured by means of Renyi entropy.

13.3 Results

13.3.1 Description of Data and Computer Experiments

In our numerical experiments, the following datasets were tested:

- 'Lenta' dataset (from lenta.ru news agency [35]). This dataset contains 8,630 documents with a vocabulary of 23,297 unique words in the Russian language. Each of these documents is manually assigned with a class from a set of 10 topic classes. However, some of these topics are strongly correlated with each other. Thus, the documents in this dataset can be represented by 7–10 topics.

- '20 Newsgroups' dataset [36]. This dataset consists of 15,404 news articles with 50,948 unique words. Each of the news items is assigned to one or more of 20 topic groups. Since some of these topics may be combined, 14–20 topics can represent the documents of this dataset [37].

In order to determine the influence of regularization on TM we investigated the models, which were discussed in section 2.1, namely: (1) pLSA model [19]; (2) LDA GS model [20]; (3) VLDA model [1]; (4) BigARTM model [10]. Additionally, we compared the results of the Renyi entropy approach for determining the 'optimal' number of topics with the results of HDP model. In our numerical experiments the number of topics T was varied in the range [2;50] in the increments of one topic. For LDA GS model, hyper-parameters α and β were varied in the range [0.1;1] in the increments of 0.1. For BigARTM model we used the following values of τ: 0.01, 0.1, 1, and 10. For each topic model and for each dataset we calculated log-likelihood and Renyi entropy.

Let us note that computational efficiency of Renyi entropy approach turned out to be much higher than that of log-likelihood. For instance, calculation of Renyi entropy for the Lenta dataset under variation of T in the range [2;50] in the increments of one took about 15 min, while calculation of log-likelihood for the same data took about nine hours. Such a great difference occurs because for Renyi entropy calculation it is enough to scan matrix Φ once, while for log-likelihood calculation one needs to multiply components of two large matrices (Φ and Θ). The purpose of our experiments was, firstly, to confirm that Renyi entropy allows us to determine the 'optimal' number of topics for the above datasets and to compare the results of this approach with the results obtained by HDP model. Secondly, the purpose was to estimate the influence of hyper-parameters on results of TM and to specify which variant of regularization gives better results according to log-likelihood and Renyi entropy.

13.3.2 Optimal Number of Topics: HDP vs Renyi Entropy in LDA GS, VLDA and pLSA

To compare the results of HDP model, pLSA, VLDA and LDA GS, we calculate weights of topics for HDP model, and Renyi entropy for pLSA, VLDA and LDA GS. In this experiment, we used the software (available

at `https://github.com/chyikwei/bnp`) which implements the online variational Bayes for the HDP proposed in work [22] and is optimized with cython. This algorithm was developed to analyze large datasets and is essentially faster than traditional algorithms [21,38].

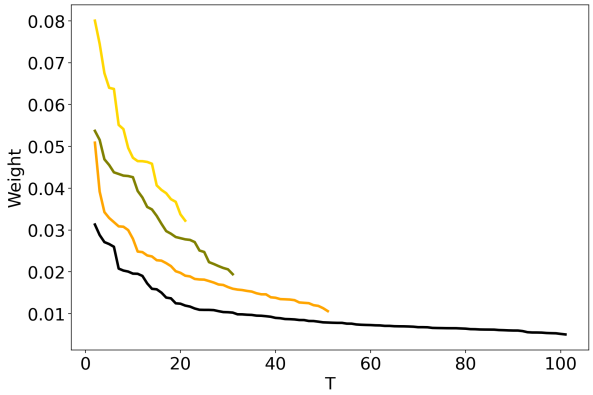


Figure 51: Distribution of weights over the number of topics T for HDP model (Lenta dataset). TLT (100)—black, TLT (50)—orange, TLT (30)—olive, TLT (20)—gold.

Figure 1 plots together the outputs of four solutions of HDP model (Lenta dataset) that differ by the values of top-level truncation parameter (TLT): 100, 50, 30, and 20. Following [39], each output is represented by a curve which sorts the weights of all inferred topics (whose number is always equal to TLT) in a descending order. The idea is to give the user an opportunity to cut off low-weight topics and to postulate that the "true" number of topics is equal to the number of high-weight topics. However, as can be seen, there is no clear threshold between high-weight and low-weight topics. The curves are monotone decreasing and do not allow to define the optimal number of topics. The same result was obtained for the 20 Newsgroups dataset. Moreover, we applied the method proposed by Wang and Blei [40] on both Russian and 20 Newsgroups corpora. This method proposes a truncation-free stochastic variational inference algorithm for HDP, which adapts the model complexity on the fly instead of requiring truncation values. For 100 runs, the method consistently inferred 28 topics on 'Lenta' corpus and 24 topics on 20 Newsgroups corpus with default parameters. Recent progress in the inference algorithms of Bayesian nonparametric models was made in work [41] which provides promising results in terms of speed and quality. However, to the best of our knowledge, this algorithm was only applied to the tasks of image categorization but not topic modeling so far.

Figure 2 demonstrates Renyi entropy curves calculated according to (2) for three topic models (pLSA, LDA GS with $\beta = 0.1$, $\alpha = 0.5$ and VLDA). For VLDA model, the number of topics was varied while the hyper-parameter α was selected automatically during the modeling. One can see that all three

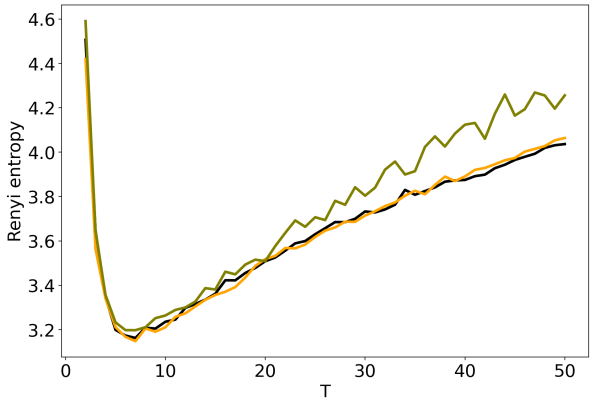


Figure 52: Distribution of Renyi entropy over the number of topics T (Lenta dataset). pLSA—black, LDA GS ($\beta = 0.1$, $\alpha = 0.1$)—orange, VLDA—olive.

curves have explicit minima of entropy. Moreover, entropy curves are very similar and the locations of minima are almost identical, namely, 7–8 topics. We obtain analogous results for the 20 Newsgorups dataset. Therefore, we conclude that Renyi entropy allows us to determine the 'optimal' number of topics for LDA GS, VLDA and pLSA models and this number is close to the human mark-up.

13.3.3 Influence of Hyper-Parameters: pLSA vs LDA GS Model

Let us discuss the influence of hyper-parameters α and β of LDA GS model on results of TM. Figure 3 demonstrates dependence of log-likelihood on the number of topics for different values of α and β (Lenta dataset). One can see that the increase in the values of hyper-parameters leads to the decrease in log-likelihood, which means that the model deteriorates as values of hyper-parameters increase. For $\alpha = \beta = 1$ we obtain the worst result. However, these curves do not allow us to determine simultaneously the optimal values of regularization parameters and the optimal number of topics. The behaviour of log-likelihood for these models on 20 Newsgroups dataset is similar to that for the Lenta dataset and, therefore, we do not provide figures.

Figures 4 and 5 plot the curves of Renyi entropy for pLSA and LDA GS with different values of hyper-parameters. One can see that the increase in the values of hyper-parameters lifts the entire entropy curve, i.e., entropy increases on average. According to the entropy approach, the best model is the model with minimum entropy. It follows that the optimal models among the considered ones are pLSA and LDA GS with $\alpha = 0.1$, $\beta = 0.1$. Notice that minima of these optimal models coincide. Numerical experiments demonstrate that minimal values of Renyi entropy for Lenta dataset are

obtained with the following combinations of model parameters: (1) $T = 7$, $\beta = 0.1$, $\alpha = 0.1$; (2) $T = 9$, $\beta = 0.1$, $\alpha = 0.5$; (3) $T = 14$, $\beta = 1$, $\alpha = 1$. Analogously, for 20 Newsgroups dataset, the minima of Renyi entropy correspond to the following combinations of parameters: (1) $T = 17$, $\beta = 0.1$, $\alpha = 0.1$; (2) $T = 15$, $\beta = 0.1$, $\alpha = 0.5$; (3) $T = 13$, $\beta = 1$, $\alpha = 1$. Instability of TM leads to the fact that entropy minimum can be determined only with the accuracy up to ± 3 topics [7]. Therefore, it makes more sense not to determine the exact minimum but to search for the location of a trough. Let us notice that values $\alpha = 1$, $\beta = 1$ lead not only to the growth of the entropy values on average but also to the horizontal shift of the minimum. One can conclude that the optimal values of hyper-parameters for LDA GS model with respect to Renyi entropy are $\alpha = 0.1$, $\beta = 0.1$. It follows that Renyi entropy approach allows us to determine both the optimal values of hyper-parameters and the optimal number of topics, while log-likelihood metric allows us to determine the optimal values of hyper-parameters only.

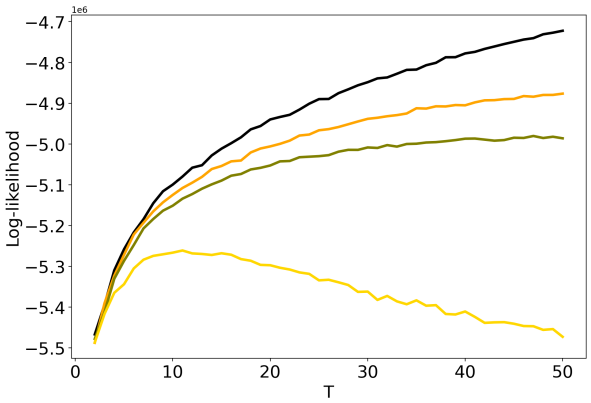


Figure 53: Dependence of log-likelihood on the number of topics T for different α and β (Lenta dataset). pLSA - black, LDA GS ($\alpha = 0.1$, $\beta = 0.1$)—orange, LDA GS ($\alpha = 0.5$, $\beta = 0.1$)—olive, LDA GS ($\alpha = 1$, $\beta = 1$)—gold.

13.3.4 Influence of Regularization Coefficients: BigARTM vs pLSA

We further discuss the influence of regularization parameters of BigARTM model on the results of TM. Here we consider sparsing regularizers for matrix Φ (sparse phi) and matrix Θ (sparse theta), where τ is regularization coefficient. Figures 6 and 7 show the behavior of log-likelihood under variation of the number of topics for different values of regularization coefficients. Both figures show that the increase in regularization coefficient impairs the model. The same result is obtained for the 20 Newsgroups dataset. Let us note that the curve of log-likelihood does not allow us to understand what happens with TM if one changes regularization coefficient and the number

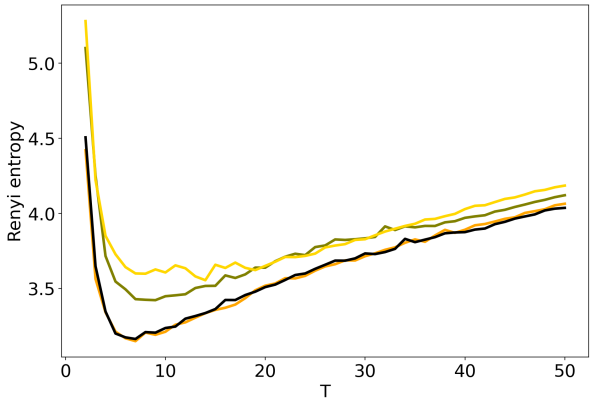


Figure 54: Dependence of Renyi entropy on the number of topics T for different α and β (Lenta dataset). pLSA—black, LDA GS ($\alpha = 0.1$, $\beta = 0.1$)—orange, LDA GS ($\alpha = 0.5$, $\beta = 0.1$)—olive, LDA GS ($\alpha = 1$, $\beta = 1$)—gold.

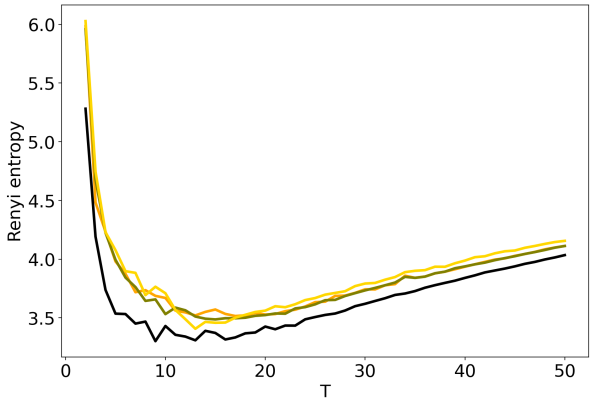


Figure 55: Dependence of Renyi entropy on the number of topics T for different α and β (20 Newsgroups dataset). pLSA—black, LDA GS ($\alpha = 0.1$, $\beta = 0.1$)—orange, LDA GS ($\alpha = 0.5$, $\beta = 0.1$)—olive, LDA GS ($\alpha = 1$, $\beta = 1$)—gold.

of topics simultaneously.

Figures 8 and 9 plot Renyi entropy curves for BigARTM model, which was run on the Lenta dataset under variation of the number of topics for different values of regularization coefficient. One can see that the range of coefficients [0.01; 0.1] gives small fluctuations in entropy minimum. In addition, these minima are located in the range [7; 10] which corresponds to the human mark-up for this dataset. However, regularization coefficient $\tau = 1$ leads to significant distortion of the Renyi entropy curve, i.e., to the lift of the entire curve and to the shift of the Renyi entropy minimum. This behavior is similar to that observed in Figures 4 and 5 for hyper-parameters of LDA GS.

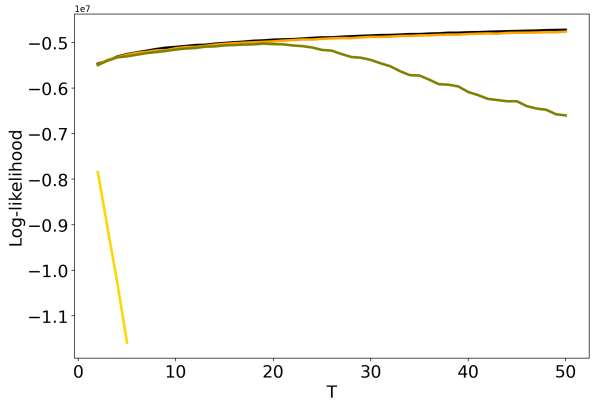


Figure 56: Dependence of log-likelihood on the number of topics T for different sparse phis (Lenta dataset): 1. pLSA—black. 2. BigARTM sparse phi ($\tau = 0.01$)—orange. 3. BigARTM sparse phi ($\tau = 0.1$)—olive. 4. BigARTM sparse phi ($\tau = 1$)—gold.

Likewise, the behavior of Renyi entropy for BigARTM on the 20 Newsgroups dataset (Figures 10 and 11) is identical to that for the Lenta dataset: the curve gets distorted when $\tau = 1$. The minimum of Renyi entropy correposnds to $T = 10$ for $\tau = 1$ in Figure 10. Additionally, in both datasets the distortion introduced by regularizing Φ is visibly larger than the effect of Θ. Our experiments show the existence of a trade-off between model quality as determined by Renyi entropy, and regularization that allows to obtain e.g. sparse or smooth topics. In BigARTM, the smallest distortions are observed with the smallest τ which yields solutions close to the entirely unregularized model—pLSA. A similar result was obtained in [42], where pLSA was shown to perform better than any regularized BigARTM model, except the one with a dictionary-based regularizer.

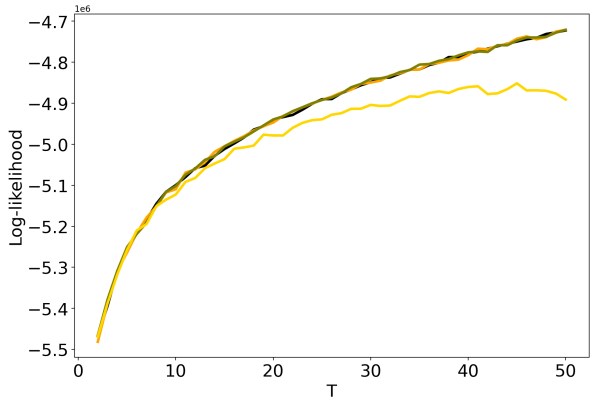


Figure 57: Dependence of log-likelihood on the number of topics T for different sparse thetas (Lenta dataset): 1. pLSA—black. 2. BigARTM sparse theta ($\tau = 0.01$)—orange. 3. BigARTM sparse theta ($\tau = 0.1$)—yellow. 4. BigARTM sparse theta ($\tau = 1$)—gold.

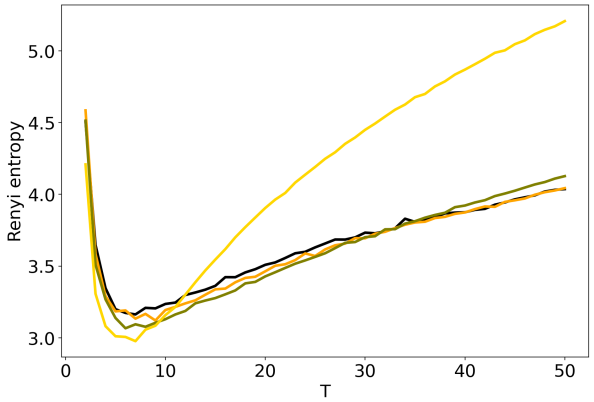


Figure 58: Dependence of Renyi entropy on the number of topics T for different sparse phis (Lenta dataset): 1. pLSA—black. 2. BigARTM sparse phi ($\tau = 0.01$)—orange. 3. BigARTM sparse phi ($\tau = 0.1$)—olive. 4. BigARTM sparse phi ($\tau = 1$)—gold.

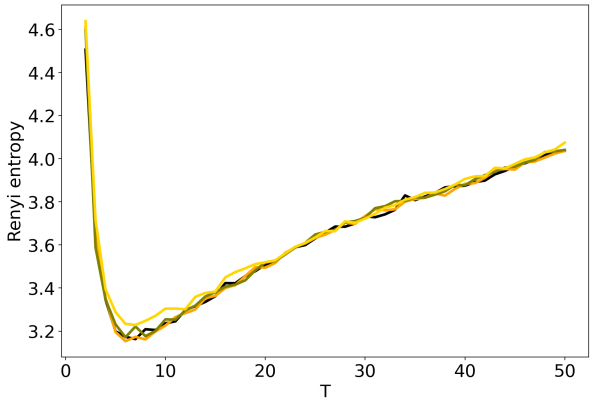


Figure 59: Dependence of Renyi entropy on the number of topics T for different sparse thetas (Lenta dataset): 1. pLSA—black. 2. BigARTM sparse theta ($\tau = 0.01$)—orange. 3. BigARTM sparse theta ($\tau = 0.1$)—olive. 4. BigARTM sparse theta ($\tau = 1$)—gold.

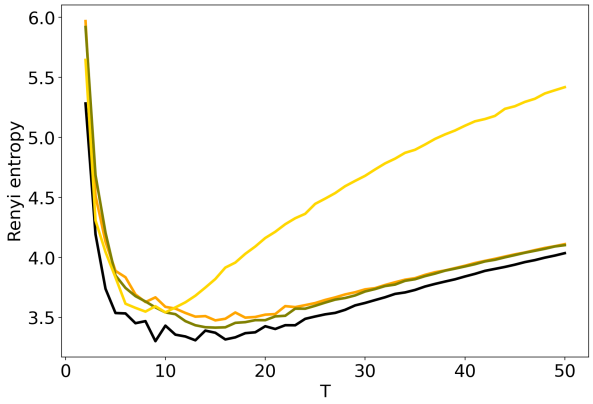


Figure 60: Dependence of Renyi entropy on the number of topics T for different sparse phis (20 Newsgroups dataset): 1. pLSA—black. 2. BigARTM sparse phi ($\tau = 0.01$)—red. 3. BigARTM sparse phi ($\tau = 0.1$)—green. 4. BigARTM sparse phi ($\tau = 1$)—blue.

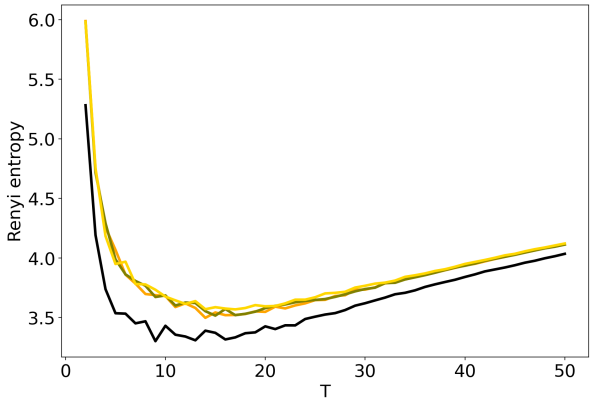


Figure 61: Dependence of Renyi entropy on the number of topics T for different sparse thetas (20 Newsgroups dataset): 1. pLSA—black. 2. BigARTM sparse theta ($\tau = 0.01$)—red. 3. BigARTM sparse theta ($\tau = 0.1$)–green. 4. BigARTM sparse theta ($\tau = 1$)—blue.

13.4 Discussion

We have proposed a method based on Renyi entropy for estimating the influence of model hyper-parameters and of regularization on the results of TM. This method was tested on pLSA, LDA GS, VLDA and BigARTM models. We demonstrated that higher levels of regularization and higher values of hyper-parameters lead to lower log-likelihood and higher entropy which is a clear sign of model deterioration. They also shift the minimum of Renyi entropy away from the optimal number of topics as determined by human mark-up. However, since both metrics indicate the highest model quality there where the values of α, β and τ are low, Renyi entropy (unlike log-likelihood) may be used not only for finding the optima of those values, but also for finding an optimal number of topics, since it is in the range of low α, β and τ that Renyi entropy performs most accurately. In addition, calculation of Renyi entropy is simpler and faster than calculation of log-likelihood. Meanwhile, HDP does not provide clear thresholds to select the optimal number of topics. We conclude that Renyi entropy can be effectively used for estimating the influence of regularization coefficients and hyper-parameters on the results of TM, determining the optimal number of topics and estimating the effect of distortion under the condition of simultaneous change of multiple model parameters.

However, our work has some limitations. First, we test our approach only on two datasets in European languages. We would like to mention that these datasets were selected since they have manual markup, therefore, they can be used as 'gold standard' datasets for testing. It would be useful to test this approach on other datasets in different languages even if they are not marked up. Second, our approach does not take into account the

quality of topic solutions in the sense of semantic stability. However, it is known that regularization may lead to an increase in the stability of TM [43] that is essential for end-users of TM. This observation may lead to further development of the model parameter selection principle and deserves a separate paper.

Author Contributions: Conceptualization, S.K. and S.S.; methodology, S.K.; software, S.K. and Z.B.; validation, S.K., Z.B. and V.I.; formal analysis, S.K.; investigation, S.K. and V.I.; resources, S.K.; data curation, S.K.; writing–original draft preparation, S.K., V.I. and Z. B.; writing–review and editing, S.S., V.I and Z.B; visualization, S.K. and V.I.; supervision, S.K. and S.S.; project administration, S.K.; funding acquisition, S.K. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: Sergei Koltcov and Vera Ignatenko were supported by the Basic Research Program at the National Research University Higher School of Economics in 2019. Zeyd Boukhers and Steffen Staab were previously supported by the German Research Foundation (DFG) through the project grant 'Extraction of Citations from PDF Documents (EXCITE)' under grant number STA 572/14-1. Steffen Staab is now supported by the German Research Foundation (DFG) through the project grant "Open Argument Mining" (grant number STA 572/18-1).

14 Paper 15: Topic Modeling

EXCITE – A toolchain to extract, match and publish open literature references

Azam Hosseini, Behnam Ghavimi, ***Zeyd Boukhers****, and Philipp Mayr*

(DOI: 10.1109/JCDL.2019.00105)

Abstract This demo paper presents a generic toolchain to extract, segment and match literature references from full text PDF files in the project EXCITE. The aim of EXCITE is extracting and matching citations from social science publications and making more citation data available to researchers. Each single step in the EXCITE pipeline and the open source tools used to accomplish the tasks are explained. The public demo system which integrates all components of the toolchain under an user-friendly interface is put forward and illustrated. As a final step, a special component is introduced which is capable to ingest the extracted and matched references into the Open Citation Corpus.
Keywords: *Reference Extraction, Reference Matching, Open Citations*

14.1 Introduction

Despite the widely acknowledged benefits of citation data, the open access to references/citations is still insufficient. Some commercial companies such as Clarivate Analytics, Elsevier or Google possess citation data in large-scale and use them to provide services for their users. On the other side, the shortage of citation data for the international and German social sciences is well known to researchers in the field and has itself often been subject to academic studies [241]. The accessibility of information in the social sciences lags behind other fields (e.g. the natural sciences) where more citation data is available.

Recently, some initiatives and projects e.g. the "Open Citations" project or the "Initiative for Open Citations" focus on publishing citation data openly[59]. The "Extraction of Citations from PDF Documents" – EXCITE[60] project is one of these projects. The aim of EXCITE is extracting and matching citations from social science publications [201] and making more citation data available to researchers. EXCITE is focusing on social science publications in German language but is introducing a generic toolchain which can be used and trained for any domain. All tools in the EXCITE project are made available to other researchers. This demo paper introduces the EXCITE toolchain.

[59] `https://i4oc.org/`

[60] `http://excite.west.uni-koblenz.de/website/`

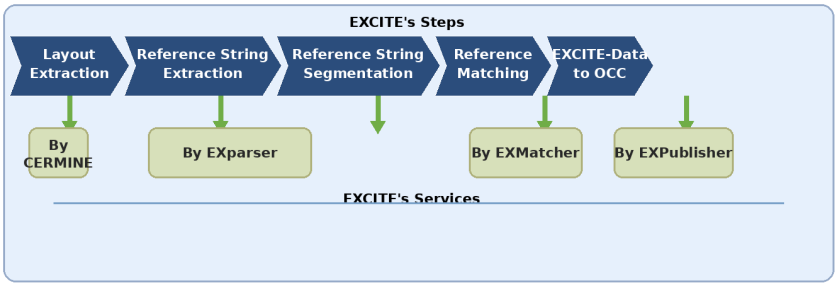


Figure 62: An overview of processing steps and tools in the project EXCITE

14.2 EXCITE Toolchain

A number of algorithms are developed in the EXCITE project for extracting references from PDF full texts and matching them against bibliographic databases (see overview in Figure 1). The extraction of references is implemented as a four-steps process:

1. Extraction of text from PDF files by CERMINE[61],
2. Identification of reference strings and segmentation of references into its constituent fields such as author, title, etc. by Exparser [46][62],
3. Matching of references against bibliographic databases by EXmatcher [124][63],
4. Export and publication of references to reusable formats by conversion of the generated reference information to the json format with OCC ontology [85].

For the matching task in EXCITE, different target databases are utilized: a) sowiport [153], b) GESIS Search[64] and c) Crossref[65]. The EXCITE corpus (PDF files to be processed in the EXCITE project) contains SSOAR[66] documents (approx. 35k), Springer Online Journals collection (approx. 80k), and sowiport full text papers (approx. 116K). The extracted citation data from the EXCITE corpus will be integrated into GESIS Search and OCC (OpenCitations[67] Corpus). EXCITE toolchain is not depended to any citation style or language but the current system is trained by using the manually assessed EXCITE gold standard[68] (including German and English languages). The

[61] https://github.com/CeON/CERMINE
[62] https://github.com/exciteproject/Exparser
[63] https://github.com/exciteproject/EXmatcher
[64] https://search.gesis.org/
[65] https://search.crossref.org
[66] https://www.ssoar.info
[67] http://opencitations.net
[68] https://github.com/exciteproject/EXgoldstandard

EXCITE toolchain code are openly available and can be adapted to new domains and languages easily.

14.3 Demo System

The EXCITE demo system `http://excite.west.uni-koblenz.de/excite` is a web interface which is deployed to integrate different parts of the EXCITE toolchain. The used web framework is Flask which integrates Python modules within web functionality such as RESTful web service. For delegating long lasting tasks, Celery is used in the EXCITE web architecture. Celery is a task queue based on distributed message passing. It enables systems to process batch jobs in the way that each defined worker performs a task in the queue and when the task is done next one will be picked. Codes in other programming languages (e.g., CERMINE in JAVA) with standard I/O format can be easily executed by a Python module. Therefore, they can be integrated in a toolchain by inserting their related tasks in Celery queue. With Celery different modules of the EXCITE toolchain can be applied on a bunch of PDF files asynchronously. There are two main functions in the demo. First: uploading single PDF files; second: running the EXCITE toolchain and checking the generated results (see Figure 2).

1 Uploading files The first step is uploading files to the server. A unique random code will be generated as soon as a user submits a file. The code will be displayed on the demo page. The code also can be sent via email (if the user entered the email in a form). This code is necessary for tracking the results of the toolchain for the submitted file. Users can check the result on the demo page by follow up code. The result will be shown in separated tabs.

2 Running EXCITE toolchain After uploading a file, the "EXCITE toolchain" will be started automatically. There are three main steps in this process: *Layout extraction:* Extracting the layout from a PDF will be started by calling a Java module base on CERMINE. The output of this step will be a "Layout file" which contains text content of each PDF file and it's related layout information such as weight and height of each line. *Reference and Segment Extraction:* In this step Exparser will be called for extracting references from the layout file. Exparser is a python code based on CRF algorithm and does reference strings extraction and segmentation in one step to reduce error rate. The output will be provided in these different formats: plain text, xml and BibTex format. *Reference Matching:* In this step EXmatcher will be called for matching references against corresponding items in the defined target bibliographical databases. The input of EXmatcher is reference strings and segments generated in the previous step. The output will be matched document ids and the probability for each match. This algorithm is build based on the combination of a blocking technique (SOLR is used for indexing) and a SVM classifier. Figure 2 demonstrates the EXCITE

demo.

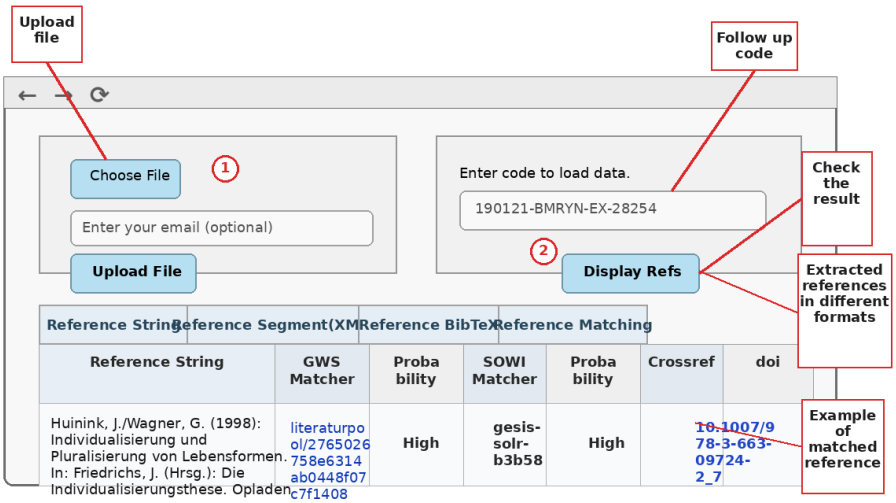


Figure 63: EXCITE Demo system

14.4 Outlook

In the remaining project time of EXCITE we will provide a Docker file containing the EXCITE toolchain to make the tools more efficiently and conveniently re-usable. The EXCITE toolchain will also be accessible as a web service API to allow third-parties to extract citation data from arbitrary publications. The EXpublisher module[69] makes sure that extracted references are enriched with the information of matched items. Afterwards, the information is converted to json format with OCC ontology and ingested into the Open Citation Corpus. OpenCitations makes this data available for users by providing dump data and also a SPARQL endpoint. Each entity (e.g., responsible agents, and reference strings) has a unique OCC identifier. For example, 'be/01101'[70] is an identifier for a reference string in OCC. The identifier contains two parts which are connected with a slash symbol. The first part, 'be' defines the type of data which is a bibliographic entity (i.e., reference string). The second part (i.e., digit part - 01101) is the main identifier. All entities in OCC with main identifier starting with '0110' are generated by the EXCITE project.

Acknowledgments

This work has been funded by Deutsche Forschungsgemeinschaft (DFG) under grant numbers MA 3964/8-1 and STA 572/14-1.

[69] https://github.com/exciteproject/EXpublisher
[70] http://opencitations.net/corpus/be/01101.html

15 Paper 16: Topic Modeling

PADME-SoSci: A Platform for Analytics and Distributed Machine Learning for the Social Sciences

***Zeyd Boukhers**, Arnim Bleier, Yeliz Ucer Yediel, Mio Hienstorfer-Heitmann, Mehrshad Jaberansary, Adamantios Koumpis, and Oya Beyan*

(DOI: 10.1109/JCDL57899.2023.00047)

Abstract Data privacy and ownership are significant in social data science, raising legal and ethical concerns. Sharing and analyzing data is difficult when different parties own different parts of it. An approach to this challenge is to apply de-identification or anonymization techniques to the data before collecting it for analysis. However, this can reduce data utility and increase the risk of re-identification. To address these limitations, we present **PADME-SoSci**, a distributed analytics tool that federates model implementation and training. **PADME-SoSci** employs a federated strategy in which all parties collaboratively implement and deploy the model, visiting each data location sequentially for training. This enables preserving data ownership and ensures that the data remains within the control of its respective owners, while still allowing the model to be trained as if all data were in a single location. Furthermore, the results are not provided until the analysis is completed on all data locations to ensure privacy and avoid bias in the results.

Keywords: *Distributed Analytics, Data Privacy, Social Sciences, Data Science*

15.1 Introduction

Data privacy and ownership are crucial in social data science as the data often includes sensitive personal information. For example, it is expected that political survey data will typically be gathered by various parties across different groups, but sharing may not occur due to privacy considerations. As a result, each party independently analyzes their own data and only shares the analytic outcome. This approach can be limited as the aggregated outcome may not accurately reflect the entire data. An alternative solution is to mask personal and sensitive information, but this can also have limitations as masked attributes may be crucial for the analysis, and their absence can negatively impact the results.

To address this issue, we present **PADME-SoSci**[71] [358], a distributed analytics tool that enables the training of models on large-scale datasets without the need to centralize the data. **PADME-SoSci** trains the analytic

[71] `https://padme-analytics.de/`

model incrementally by visiting each data station one by one. This ensures that the data remains decentralized and protected from unauthorized access. Additionally, **PADME-SoSci** only produces output after visiting all data stations, thereby avoiding the sharing of derived knowledge that could potentially reveal information about the data within a specific station. This provides an added layer of security and ensures that the data remains confidential. To further guarantee the security and privacy of the data, it is advisable that the model is implemented in a federated manner, where all involved parties participate in the implementation, code review, and deployment of the model. This ensures that the data is protected by multiple parties and that any potential vulnerabilities are identified and addressed.

15.2 PADME-SoSci

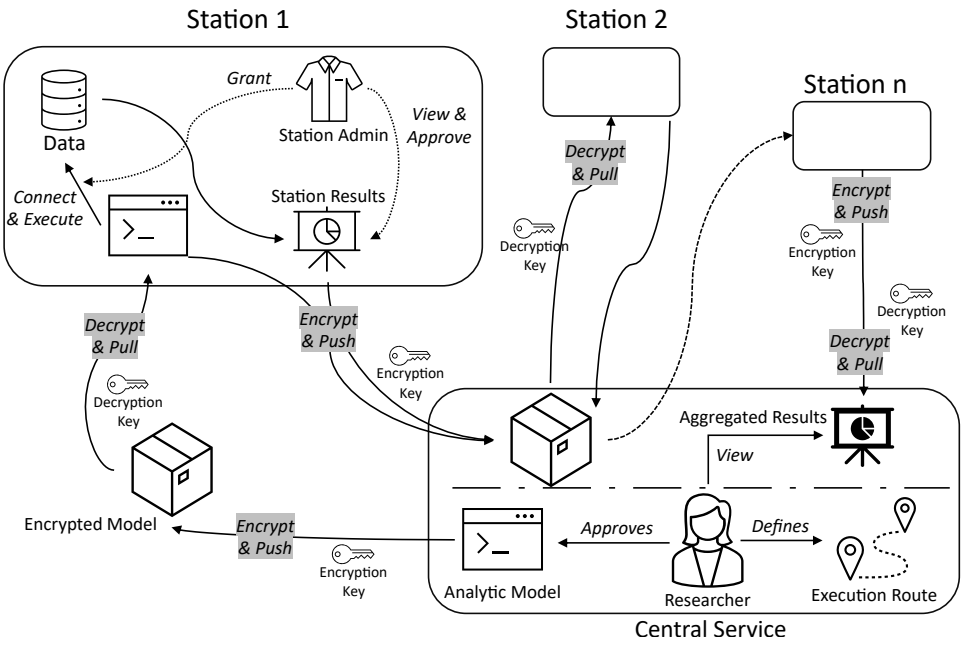


Figure 64: An overview of the distributed analytic tool with n data stations

The distributed analytics tool depicted in Figure 64 has two primary components: the *Service Center* and the *Data Stations*. The *Service Center* acts as the central hub for the entire analytics process. It oversees the development and approval of analytic models, manages the authentication and authorization of scientists and researchers, and ensures the proper execution of analytics tasks. The process of developing and approving the analytic models involves the collaboration and agreement of both the researcher and the data owners. This consensus mechanism ensures that the parameters and architecture of the global model are agreed upon by all parties. The *Service*

Center also ensures the security of the analytics process by securely packaging the analytics model into containers and managing the execution routes through authentication. This minimizes the risk of unauthorized access or misuse of the analytic models. Additionally, the *Service Center* employs encryption techniques to protect the privacy and ownership of data throughout the entire process. The encrypted containers carrying the analytics models are then transferred to the next target station.

The *Data Station* operates independently of the *Service Center*, but it remains closely connected to it through a secure communication channel. The station pulls the encrypted container from the *Service Center* after authenticating itself. The station admin then provides the necessary connection information to allow the decrypted container to execute the analytics model. The station's unique key is a crucial security feature that ensures that only the designated station can execute the analytic model within the decrypted container. After the analytics task is completed, the Station Admin reviews the results and, upon approval, packages them into a secure container using the station's specific key. This process is repeated for each station in the route until the final results are available in an encrypted container. The authenticated scientist can then retrieve, decrypt and view the aggregated results.

15.2.1 Prerequisites

To effectively utilize the distributed analytic tool, compliance with the following prerequisites is crucial:

- **Data standardization:** All used data must be in a standardized format that the model is designed to work with.
- **Distributed model:** The analytic model must be capable of being distributed. For example, the Latent Dirichlet Allocation (LDA) model requires the entire vocabulary of the corpus before running, making it unsuitable for direct use with PADME-SoSci.
- **Computational resources:** Every data station needs to have adequate computational resources to execute the analytic model.

15.3 Use Cases

15.3.1 Sentiment Analysis

Online reproducibility services, such as mybinder.org, have become popular in the Computational Social Science community [42]. These services allow researchers to share complex computational analysis pipelines in the form of notebooks that can be easily executed in a browser without the need for additional software installation. However, the pipelines are currently limited

to public data. In this paper, we demonstrate the use of PADME-SoSci for a sentiment analysis test within the following prototypical workflow[72]:

1. Schema data is created and publicly shared for a sensitive dataset that can not be shared.
2. An interested researcher that would like to access the sensitive dataset develops an analysis using the publicly available schema data.
3. The researcher submits the analysis to PADME-SoSci to execute the analysis of the private data.
4. An exit control on the analysis results is performed, and the results of the analysis are sent back to the researcher if the exit control is passed.

The publicly available part of the German Federal Election 2017 Twitter dataset[328] (DBK: ZA6926) is used for this use case.

15.3.2 Supervised Author Name Disambiguation

This demonstration showcases the capability of PADME-SoSci to distribute the Author Name Disambiguation (AND) task across two separate stations, each holding part of the data. The data used for AND is typically open, but the purpose of this demonstration is to highlight the versatility of PADME-SoSci in handling various types of data and analytical models. In particular, we are utilizing a supervised neural network model [47] that has been trained on a preprocessed DBLP dataset[73]. The DBLP dataset is split and distributed between the two stations for this demonstration.

Acknowledgement

This joint work received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number: NFDI4DataSciene[74] (*460234259*).

[72] Originally discussed on GitHub `https://github.com/gesiscss/btw17_sample_scripts/issues/4`

[73] `https://doi.org/10.5281/zenodo.7506562`

[74] `https://gepris.dfg.de/gepris/projekt/460234259`

16 Paper 17: LLM for FAIR Assessment

FAIR Data Assessment Using LLMs: The Fair-Way

Anmol Sharma, Sulayman K. Sowe, Soo-Yon Kim, Sayed Hoseini, Fidan Limani, ***Zeyd Boukhers****, Christoph Lange, and Stefan Decker*

(DOI: 10.1145/3746252.3760811)

Abstract As part of modern research practices, the FAIR data principles have become essential for data discoverability, usability, and sharing. Existing implementations for automatically assessing FAIR adherence (FAIRness) often suffer from limited usability, inconsistent accuracy, and difficult-to-interpret results, as they require explicit rules to cover for specific FAIR assessment frameworks, which are not easy to generalize. This paper introduces Fair-Way, an open source tool that leverages Large Language Models (LLMs) to automate FAIRness assessment. Fair-Way applies a divide-and-conquer approach to decompose the assessment process into fine-grained tasks, as well as to split the metadata into manageable chunks. Evaluation demonstrates that Fair-Way achieves performance comparable to existing tools, while outperforming them in several key metrics. Moreover, Fair-Way generalizes across FAIR assessment indicators without requiring explicitly programmed logic and supports both structured and unstructured metadata in diverse formats. Finally, it enables user-defined, domain-specific tests, which are typically not supported by other systems. Overall, Fair-Way represents a scalable and flexible solution to accelerate FAIR data practices across research domains.

Keywords: *FAIR Assessment, Large Language Models, FAIRification, Research Data Management*

16.1 Introduction

Given the exponential growth in research data volume, the Findable, Accessible, Interoperable, and Reusable principles (FAIR) [359] are crucial for enhancing their discoverability, reusability, and the machine-actionability [343]. Consequently, many initiatives across domains adopt these guiding principles, such as demonstrated in the works on FAIR compliance in agriculture [282], FAIR in health [134], FAIR for AI research [166], or FAIR for LLM training data [297]. In each of these cases, the principles are tailored both in scope (supporting only certain FAIR aspects which are deemed to be of higher importance than others) and in context (i.e., to a specific case or domain). Yet, evaluating the extent to which research data aligns with FAIR principles generally remains a challenge [195, 330, 61].

Manual assessment of FAIRness is time-consuming, subjective, and often inconsistent across domains [119]. Furthermore, existing automated assess-

ment tools are relatively limited in terms of the metadata types they handle, and information extraction rules they provide [330], leading to inconsistent results for sources with different characteristics from varying domains. Studies on FAIR evaluation tools [195, 61] underscore the need for a more comprehensive solution, capable of handling different types of sources, including structured and unstructured metadata, a multitude of metadata formats, and excellent information extraction abilities. Furthermore, enabling the metadata assessment prior to its publication results in a FAIRer dataset, a feature missing from automated FAIR assessment tools.

Recent developments in natural language processing and the emergence of Large Language Models (LLMs) have opened new avenues for simplifying data interactions [58]. LLMs have demonstrated remarkable capabilities in understanding and generating human-like text, making them suitable for various applications [68, 7, 36], including data management [213]. LLMs show outstanding capabilities to execute complex instructions, act as agents, generate actionable recommendations, and handle diverse information formats to gain insights into data and metadata.

In this paper, we introduce an LLM-based approach for automatic FAIR assessment, as well as the corresponding tool, *Fair-Way*, capable of handling, processing, assessing, and creating different types of metadata. Fair-Way supports the assessment of published and unpublished datasets and reduces the reliance on explicitly programmed implementation for FAIR assessment, giving it a wide application range of domains and accelerating the FAIRification process. Additionally, Fair-Way enables simple domain-specific user-provided tests for vocabularies and domain-specific standards, which is difficult to assess with existing automated tools.

In summary, we make the following contributions:

- We present a novel approach for the adoption of LLMs for automatic FAIR assessment.
- We develop Fair-Way, a tool based on this approach, that supports multiple metadata formats for both published and unpublished data, including domain-specific assessments through user-provided test prompts. Fair-Way's code, prompts, and evaluation datasets are available on GitHub and a demo showcasing the application working can be accessed here.
- We conduct an evaluation of open and closed-source LLMs for FAIR assessment tasks.

The rest of this paper is structured as follows: Section 16.2 reviews related work, Section 16.3 describes our methodology and the implementation of the tool, Section 16.4 presents an analysis of the system's capabilities and limitations, and Section 16.5 summarizes key contributions and future directions.

Type	Tool	Open Source	Metric
Self-Assessment	ARDC	–	–
Self-Assessment	EUDAT	–	–
Semi-Automated	FAIR-Shake	Yes	FAIRness Maturity Indicators
Automated	FAIR Evaluator	Partially	FAIRness Maturity Indicators
Automated	F-UJI	Yes	FAIRsFAIR
Automated	FAIR-Checker	Yes	RDA FAIR Maturity Model

Table 54: Comparison of existing FAIR assessment tools.

16.2 Related Work

FAIR assessment utilizes frameworks that define indicators and associated tests to measure a resource's adherence to FAIR principles. Prominent examples include the Research Data Alliance's FAIR Data Maturity Model (DMM) [298], the FAIRsFAIR Data Object Assessment Metrics (DOAM) [93], and the FAIRness Maturity Indicators (MIs) [360]. These frameworks provide community-agreed metrics that FAIR assessment tools evaluate.

Building on these community metrics, various tools have been developed for FAIR assessments, ranging from manual questionnaires to fully automated systems [119], as summarized in Table 54. Manual questionnaire-based tools, such as those by ARDC [25] and EUDAT [178], rely on self-assessment and promote understanding, but are time-consuming, require prior FAIR knowledge, and lack quantitative rigor. Semi-automated tools address some of the limitations mentioned above, by allowing some level of automated evaluation; FAIRshake [1, 79], for example, uses MIs and allows users to define custom assessment "rubrics". For fully automated assessment, we have tools like F-UJI [108, 94], based on the DOAM metrics, which checks for domain-agnostic indicators. However, extending it to new terminologies, metadata formats, or domain-specific checks necessitates programming and modifications to its source code. Another automated tool, FAIR-Checker [38, 24], employs the DMM metrics, using Semantic Web technologies like knowledge graphs for metadata analysis, but designed specifically for bio-informatics artifacts. While existing automated tools mark progress, they face challenges in generalizing across diverse metadata standards, languages, and formats. They require explicit programmed logic for each indicator test and resource type [94, 24]. This inherent inflexibility, including customization calls for broader applicability and the need for a more generalizable solution.

16.3 Methodology

To enable reliable and automated FAIR data assessment using LLMs, our methodology combines a FAIR assessment framework with an LLM-based processing pipeline. For this, it was crucial that the FAIR assessment framework includes practical, automated tests for every metric. Thus, we chose FAIRsFAIR DOAM [93] as, besides supporting automated tests, they are domain-agnostic, community driven, well documented and used by assessment tools like F-UJI.

To automate evaluation of our chosen metrics, we adopted a design science approach to develop and refine an LLM-driven pipeline. Initial experiments employed open source LLMs, like Microsoft Phi-4 (14b), Meta Llama3.1 (8b), and Qwen2.5-Coder (14b) [4, 225, 167]. The models were prompted with simple instructions plus full dataset metadata to perform a test evaluation on. We also specified a required JSON response format for each test to enable complete automation. Such an example instruction is *"Check if metadata includes descriptive elements, like creator, title, publisher, or publication date."*, for which the model was asked to execute the given instructions and respond strictly in the provided JSON schema.

However, this approach resulted in many inconsistent and non-reproducible outputs, as part of model hallucinations and failure to adhere to instructions, especially with smaller models that often returned lengthy explanations instead of the requested output. This can be attributed to (1) a lack of specialized fine-tuning of open source models for metadata evaluation, (2) LLM context window limitations, as performance can significantly degrade with increased token count, (3) tokenization used in models, which typically treats a word plus its space as one token, impacting how a full prompt is tokenized, and (4) structured metadata (e.g., JSON) disproportionately expanding token counts. For instance, using the prompt above on metadata retrieved from this Zenodo record into a JSON format resulted in **1822** tokens using GPT-4o tokenizer[75] and it further inflates with few-shot examples on the given task. For our solution, to mitigate these challenges, we adopted a *Divide-and-Conquer* strategy, constraining the context window of LLMs via specific prompts and processing metadata in manageable chunks rather than complete sets.

The *system's operational flow* begins with user input – either a resource URL (from supported repositories) or a local metadata file, and optional user defined domain-specific tests. Fair-Way then extracts metadata from online resources (via embedded content and repository APIs like Zenodo [105]) or by parsing uploaded files. The complete metadata is potentially chunked if determined to be large based on pre-defined metadata limits for each format, and subsequently assessed for each FAIR metric, where the LLM processes each chunk using tailored prompts and few-shot examples. Finally, successful

[75] `https://platform.openai.com/tokenizer`

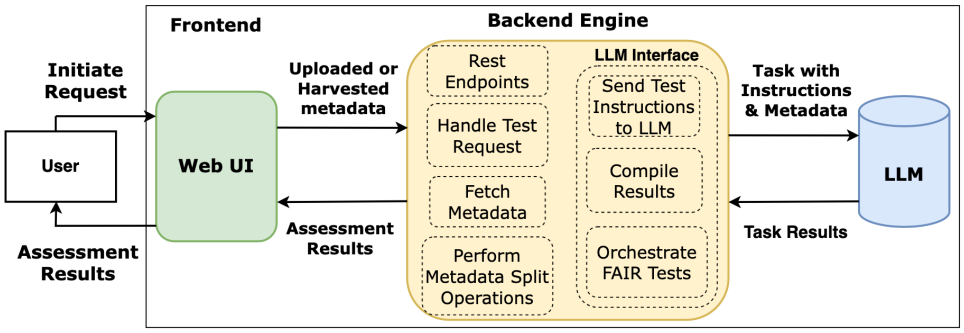


Figure 65: System Architecture

chunks for a task are combined together to form a single assessment result. The key steps of the workflow include:

- **Metadata pre-processing** reduces tokens by pruning redundant whitespaces and validating structured metadata for syntax.
- **Chunking** divides large metadata files into smaller parts using the LangChain framework to limit tokens per LLM call, while small files are processed as whole. The LLM later synthesizes results from multiple chunks for a test into single final result.
- **Task decomposition** breaks test instructions into simpler, clear prompts for the LLM, each requiring a JSON response.
- **Result reuse** for certain metrics. Tests are performed on results obtained from other tests (e.g., using extracted file information to check for open data formats).

Fair-Way's *architecture* comprises of three core components (as shown in Figure 65): **Frontend** developed with VueJS for user interaction. **Backend** is a FastAPI-based RESTful service as the main engine encompassing multiple core components. **LLM Interface** is built within the backend, for managing requests to the chosen LLM service (Ollama [2] or the OpenAI) and compiling results. The inner workings of all the components are described in detail on our GitHub repository.
The *prompts* emulate the instructions of the chosen DOAM metric specification. Based on initial observations, we iteratively curated our prompts employing several prompt engineering techniques: **Specific Instructions** with clear and precise directives for each assessment test [60]. **Structured Output** enforces JSON model output for systematic parsing and complete automation of tests. **Chain-of-Thought** instructions are used by decomposing a test into sequential subtasks [357] to follow. **Two-Shot Prompting** examples [59] with varied inputs and expected outputs are provided for each

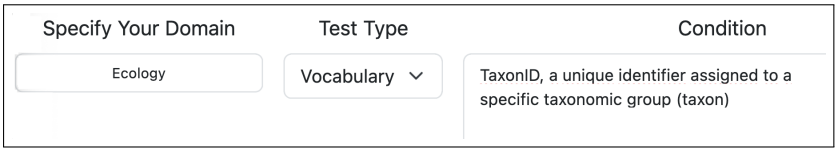


Figure 66: Domain Specific Testing

test. Our experimentation found two-shot examples as a fine balance considering context window usage aiding non-fine-tuned LLMs in each FAIR test. **Special Prompts** are also used to combine results from multiple chunks and sources into a single result for a single test. These prompting techniques, combined with the pruning and splitting strategies for metadata, enable LLMs to perform automated FAIR assessments. For *domain-specific* testing, users are asked to select a vocabulary test or a domain standard test, specify their domain, and provide relevant vocabulary terms or a standard domain check (as prompt) for evaluation by LLM. As shown in Figure 66, a user can define a vocabulary check for the Ecology domain by providing a term to check and a brief description. These are evaluated independently in addition to the domain-agnostic DOAM metrics. The user-provided tests, composed of test types, test instructions and the corresponding domains, are then passed to the LLM as part of a special prompt to perform domain-specific testing. Collectively, these functionalities support researchers in curating metadata that adheres to FAIR principles, complies with domain-specific standards, and facilitates the publication of research artifacts to promote discoverability and seamless sharing.

16.4 Evaluation & Discussion

Our evaluation is twofold: (i) identifying the most suitable LLM for FAIR assessment by comparing several LLMs, and (ii) benchmarking Fair-Way against existing automated tools with the same metrics. To select Fair-Way's optimal LLM, we evaluated GPT-4o and a few open-weights models, prioritizing performance, deployability, and structured output capabilities. We created a benchmark consisting of 15 datasets across five domains with manually curated ground truths from harvested metadata for each dataset example for each of utilized DOAM metrics. The curated ground truths are available in our (GitHub) repository. Performance was assessed using four metrics, based on a 0 (lowest) to 1 (highest) scale: **Mean Structural Accuracy** for adherence to the required JSON output schema. **Mean Sequence Match** for accuracy of extracted entity lists (1.0=exact set match, 0.5=partial overlap, 0.0=empty set intersection). **Exact Match Accuracy** for verbatim correctness of specific entities embedded in metadata. **Mean BERT Score** for semantic similarity of long texts (e.g., comments, summaries, etc.)

LLM	Temp.	Struct. Acc.	Seq. Match	Exact Match	BERTScore
GPT-4o	0.3	1.000	**0.557**	**0.905**	**0.937**
Mistral-Small	0.3	1.000	0.523	0.840	0.869
Phi-4	0.7	1.000	0.390	0.824	0.890
Llama 3.3	0.5	0.989	0.520	0.603	0.888
Qwen2.5-Coder	0.3	1.000	0.457	0.778	0.909

Table 55: Fair-Way LLM FAIR assessment comparison.

Table 55 summarizes the LLM comparison on our benchmark, showcasing best metrics per model with the optimal temperature. We standardized hyperparameters *Top-p* (0.9) and *context window* (max. 5800 tokens), varying the temperature parameter (range: 0.3–0.9) to identify optimal settings for each model. Final scores were averaged over two independent runs per dataset for each model and temperature combination to mitigate variability. OpenAI's GPT-4o achieved the highest scores across all metrics, emerging as the optimal model for FAIR assessment. Lower temperatures yielded more precise outputs for most models, excluding Phi-4 [4]; Mistral-small [175] was the top-performing open source model.

The Table **??** summarizes the metric completion comparison between Fair-Way (using GPT-4o model) and F-UJI [108] as standalone tools. The comparison used three randomly selected datasets from different domains: an (Earth Sciences) [179], a (Finance) [118], and a (Climate Sciences) [23]. We implemented 12 out of 17 FAIRsFAIR DOAM metrics (v0.5) with respective scores for each metric, focusing on those amenable to LLM-based information extraction and reasoning, excluding metrics like FsF-A1-02M (protocol accessibility) better suited for explicitly programmed logic. Suffixes D and M denote data and metadata metrics, respectively.

For *Findability*, both tools performed similarly on identifier detection (FsF-F1-01D/02D) and core metadata vocabulary checks (FsF-F2-01M). In *Accessibility*, Fair-Way identified public access conditions (FsF-A1-01M) for all datasets, while F-UJI failed for two Zenodo datasets, succeeding only in the Climate Sciences example. For *Interoperability*, both tools identified embedded structured metadata (FsF-I1-01M) however, both systems failed for the metric (FsF-I2-01M). F-UJI failed to check semantic resource namespaces and Fair-Way produced false positives. Notably, Fair-Way demonstrated superior performance in depth of information extracted for related entities (FsF-I3-01M), identifying citations, ORCID and RoR IDs, while F-UJI detected only version even though both succeeded in the test. In the *Reusability* assessment, both tools correctly identified licenses (FsF-R1.1-01M) and provenance (FsF-R1.2-01M) information. However, F-UJI provided incomplete information regarding dataset file formats (FsF-R1.3-02D) across all

three datasets, a metric Fair-Way successfully completed.

Overall, Fair-Way completed more common metrics successfully than F-UJI: 10/12 vs. 8/12 for Earth Sciences; 11/12 vs. 9/12 for Finance; and 11/12 vs. 9/12 for Climate Sciences. This demonstrates Fair-Way's performance is at least on par with, and often superior to, F-UJI, particularly on metrics like FsF-A1-01M and FsF-R1.3-02D while providing more accurate and detailed responses by extracting richer information from metadata.

16.5 Conclusion

This paper introduced Fair-Way, an open source tool leveraging LLMs to automate and improve FAIR assessment by providing detailed assessment results. Our evaluation showed that Fair-Way achieves performance comparable to, and in several aspects superior to, existing tools, particularly in its ability to handle diverse metadata formats and generalize across assessment indicators.

Despite its promising capabilities, Fair-Way has limitations. The reliance on non-fine-tuned LLMs introduces a risk of hallucination on certain metrics, potentially affecting assessment precision. Additionally, the current iteration of Fair-Way employs purely LLM-based assessment tests, although certain common metrics are easily verifiable with explicitly implemented logic. To further improve accuracy and evaluation times the system would utilize a more hybrid approach utilizing MCP and tool calling to verify results of certain metrics. Furthermore, the computational demands of LLMs can lead to slower evaluation times, especially if powerful GPU hardware is unavailable.

Future work will address these limitations and expand Fair-Way's utility by offering actionable suggestions to improve metadata FAIRness post-assessment. To mitigate hallucination and improve task-specific accuracy, fine-tuning LLMs for FAIR assessment is also a significant research direction. Building on this foundation, additional FAIR metrics – challenging for traditional logic but amenable to LLMs will also be explored. These advancements aim to further establish Fair-Way as a scalable, flexible, and comprehensive solution for promoting FAIR data practices across research disciplines.

acknowledgment

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), under the project NFDI4DataScience, grant No. 460234259

Generative AI Disclosure

During the preparation of this work, the authors used OpenAI's generative AI ChatGPT, DeepL, and Grammarly to improve the writing, make sug-

gestions, and re-phrase. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

17 Conclusion and Future Directions

This habilitation thesis has presented seventeen contributions organised around five research themes, unified by the argument that robust knowledge processing requires methods that are adaptive to domain, aware of multiple modalities, and scalable in practice. This section summarises the key findings per theme, identifies concrete future research directions that emerge from the results, and closes with a reflection on the broader research vision.

17.1 Summary of Contributions

Theme 1: Scholarly Knowledge Infrastructure (P1, P2, P3, P4, P14, P15, P16, P17). The contributions in this theme demonstrate that metadata extraction from scientific documents benefits consistently from progressively richer representations. Moving from CRF-based sequence labelling (P1) through CNN-based transfer learning (P2) to multimodal spatial-semantic encoders that jointly model text and document layout (P3, P4) yielded measurable gains across diverse document collections, including German social science publications with heterogeneous formatting. The EXCITE toolchain (P15) operationalises the reference extraction methods of P1 in a reusable end-to-end system. Complementary contributions established that topic model behaviour can be understood and controlled through Rényi entropy analysis (P14), that privacy-preserving distributed analytics is feasible for cross-institutional scholarly research (P16), and that LLM-based FAIRness assessment provides a scalable alternative to manual or rule-based evaluation (P17). Together, these contributions address the scholarly knowledge lifecycle from extraction through analysis to quality assessment.

Theme 2: Author Name Disambiguation (P5, P6). The contributions in this theme show that even minimal bibliographic metadata, specifically co-authorships, publication titles, and venue information, carries sufficient signal for effective author disambiguation using supervised learning models, though challenges remain for cases involving new co-authorships or highly ambiguous name variants. This finding, validated on DBLP data at multiple ambiguity levels, has practical implications for digital libraries where records often lack rich metadata such as abstracts or affiliations. P5 received the Best Paper Award at TPDL 2022.

Theme 3: Domain-Specific Text Classification and Prediction (P7, P8, P9, P13). In healthcare, the contributions demonstrate that external knowledge integration, whether through medical ontologies (P7) or LLM-derived representations (P8), is more decisive for ICD coding performance than model scale alone. P13 reinforces this finding by showing that smaller,

domain-specifically fine-tuned LLMs can match or exceed larger general-purpose models in clinical information extraction. In finance, P9 establishes that public awareness and trader sentiment, captured from social media and behavioural signals, provide significant predictive power for cryptocurrency volatility beyond traditional trading indicators.

Theme 4: Multimodal Interpretability (P10). The COIN method demonstrates that counterfactual image generation is a viable approach for interpreting VQA models, identifying decisive image regions through minimal modifications that change the predicted answer. The method operates without access to model internals, making it applicable as a post-hoc explanation tool for arbitrary VQA architectures.

Theme 5: LLM Optimisation (P11, P12). The EMORL framework (P11) shows that ensemble-based multi-objective reinforcement learning can balance competing fine-tuning objectives efficiently, avoiding the need for separate training runs per objective. The knowledge distillation study (P12) demonstrates that compressed student models retain over 90% of teacher performance on question-answering tasks while reducing computational requirements by up to 57%. Both contributions address the deployment gap between LLM research and practical application.

Cross-cutting findings. Beyond the theme-specific results, this thesis yields three programme-level insights. First, representation quality is consistently more decisive than architectural complexity across all domains studied. Second, the integration of heterogeneous knowledge sources, whether document layout, medical ontologies, co-authorship graphs, or social media signals, compensates for limitations of the primary textual data in every theme. Third, the progression from using LLMs as tools (Themes 1, 3) to optimising them as objects of study (Theme 5) reflects a natural and productive research trajectory that connects practical experience with methodological innovation.

17.2 Future Research Directions

The following directions emerge from specific findings, limitations, and open questions identified during the research presented in this thesis.

From document-level to corpus-level extraction. The metadata extraction methods developed in Theme 1 operate on individual documents. However, the experiments in P1 and P4 revealed that extraction errors are not uniformly distributed but cluster in documents with unusual layouts or degraded scan quality. A promising direction is to exploit corpus-level consistency, for instance by using successfully extracted records to constrain or

correct extraction in related documents from the same collection or publisher. Few-shot and in-context learning with LLMs offer a natural framework for this, building on the LLM integration explored in P8, P13, and P17.

Dynamic author profiles. The disambiguation models in P5 and P6 operate on static snapshots of bibliographic databases. In practice, author profiles evolve as researchers change institutions, shift topics, and acquire new co-authors. The current models would need to be retrained to incorporate such changes. An incremental learning framework that updates author representations as new publications appear, without full retraining, would make the approach suitable for continuously maintained digital libraries.

Temporal and longitudinal modelling for clinical coding. The ICD coding contributions (P7, P8) process individual discharge summaries in isolation. Clinical practice, however, involves longitudinal patient histories where prior diagnoses, treatments, and disease progression inform current coding decisions. Integrating temporal patient trajectories into the coding pipeline, potentially through the knowledge-augmented architectures developed in P7 combined with sequence models over patient timelines, is a concrete next step. The privacy-preserving infrastructure from P16 could enable such longitudinal modelling across institutional boundaries.

Explainability as a design principle. The interpretability work in P10 (COIN) operates post-hoc, explaining predictions after they have been made. A more ambitious direction is to incorporate interpretability as a training objective, producing models that are inherently more transparent. The multi-objective optimisation framework of P11 (EMORL) provides a natural mechanism for this: interpretability metrics could be included as one of the objectives to be balanced during fine-tuning, alongside accuracy and fluency. This would connect Themes 4 and 5 in a way that neither contribution currently achieves.

Multi-objective optimisation beyond language. The EMORL framework (P11) was evaluated on language generation tasks, but its ensemble-based hidden-state aggregation mechanism is not inherently language-specific. Applying this framework to other multi-objective settings, such as balancing extraction accuracy against processing speed in the metadata extraction pipeline of Theme 1, or balancing sensitivity against specificity in clinical coding (Theme 3), would test the generality of the approach and potentially yield practical improvements in those domains.

Cross-lingual and cross-cultural scholarly analytics. The metadata extraction work in Theme 1 addressed the challenge of German-language

publications, and the disambiguation work in Theme 2 encountered the particular difficulty of name ambiguity across cultures. A systematic investigation of cross-lingual metadata extraction and disambiguation, leveraging multilingual pre-trained models and the multimodal approach of P3 and P4 (where layout features are language-agnostic), would extend the impact of these contributions to a broader range of scholarly collections.

17.3 Final Remarks

The research presented in this thesis spans multiple domains, methods, and data modalities, yet it is held together by a single conviction: that the most productive advances in knowledge processing emerge at the intersection of domains and modalities rather than from incremental improvements within a single established setting. The progression from extracting metadata from heterogeneous documents to disambiguating the authors behind those documents, from classifying clinical and financial texts to interpreting the models that perform the classification, and from using large language models as tools to making them efficient enough for practical deployment, illustrates a research trajectory in which each step creates the conditions for the next.

Looking ahead, the convergence of large language models, multimodal learning, and domain-specific knowledge integration is reshaping the landscape of computational knowledge processing. The methods and insights developed in this thesis provide a foundation for navigating this convergence. In particular, the recurring finding that external knowledge integration and representation quality outweigh sheer model scale suggests that future systems will benefit more from thoughtful domain adaptation than from scaling alone.

As a "*Privatdozent*", my research agenda will continue to pursue the cross-domain, multimodal perspective that characterises this thesis. Three priorities will guide this agenda: first, extending the scholarly knowledge infrastructure towards fully automated, self-correcting pipelines that maintain FAIR-compliant metadata at scale; second, deepening the integration of LLMs into clinical and information science workflows with a focus on trustworthiness, explainability, and resource efficiency; and third, advancing multi-objective optimisation frameworks that allow practitioners to balance competing requirements without prohibitive computational cost. The contributions presented here demonstrate that these goals are both achievable and impactful, and I look forward to pursuing them in the years ahead.

References

[1] Fairshake. https://fairshake.cloud/metric/14/. Accessed: 2024-10-17.

[2] Ollama. https://github.com/ollama/ollama. Accessed: 2025-03-10.

[3] Grobid. 2008–2021.

[4] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024.

[5] David Aboody, Omri Even-Tov, Reuven Lehavy, and Brett Trueman. Overnight returns and firm-specific investor sentiment. *Journal of Financial and Quantitative Analysis*, 53(2):485–505, 2018.

[6] Jethin Abraham, Daniel Higdon, John Nelson, and Juan Ibarra. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3):1, 2018.

[7] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[8] Blythe Adamson et al. Approach to machine learning for extraction of real-world data variables from electronic health records. *Frontiers in Pharmacology*, 2023.

[9] Kiran Adnan and Rehan Akbar. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, 2019.

[10] M Eren Akbiyik, Mert Erkul, Killian Kämpf, Vaiva Vasiliauskaite, and Nino Antulov-Fantulin. Ask" who", not" what": Bitcoin volatility forecasting with twitter data. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 688–696, 2023.

[11] Saad Ali Alahmari. Using machine learning arima to predict the price of cryptocurrencies. *The ISC International Journal of Information Security*, 11(3):139–144, 2019.

[12] Rayner Alfred, Joe Henry Obit, Mohd Hanafi Ahmad Hijazi, Ag Asri Ag Ibrahim, et al. A performance comparison of statistical and machine learning techniques in learning time series data. *Advanced Science Letters*, 21(10):3037–3041, 2015.

[13] Dilawar Ali, Kenzo Milleville, Steven Verstockt, Nico Van de Weghe, Sally Chambers, and Julie M Birkholz. Computer vision and machine learning approaches for metadata enrichment to improve searchability of historical newspaper collections. *Journal of Documentation*, 2023.

[14] Reem K Alkhodhairi, Shahad R Aljalhami, Norah K Rusayni, Jowharah F Alshobaili, Amal A Al-Shargabi, and Abdulatif Alabdulatif. Bitcoin candlestick prediction with deep neural networks based on real time data. *CMC-COMPUTERS MATERIALS & CONTINUA*, 68(3):3215–3233, 2021.

[15] Rabah Alzaidy, Cornelia Caragea, and C Lee Giles. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The world wide web conference*, pages 2551–2557, 2019.

[16] Dong An, Liangcai Gao, Zhuoren Jiang, Runtao Liu, and Zhi Tang. Citation metadata extraction via deep neural network-based segment sequence labeling. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1967–1970, 2017.

[17] Dong An, Liangcai Gao, Zhuoren Jiang, Runtao Liu, and Zhi Tang. Citation metadata extraction via deep neural network-based segment sequence labeling. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1967–1970, New York, NY, USA, 2017. Association for Computing Machinery.

[18] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[19] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[20] Sam Anzaroot and Andrew Mccallum. A new dataset for fine-grained citation field extraction. *ICML Workshop on Peer Reviewing and Publishing Models.*, 2013.

[21] D Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[22] Tasleem Arif, Rashid Ali, and M Asger. Author name disambiguation using vector space model and hybrid similarity measures. In *2014 Seventh International Conference on Contemporary Computing (IC3)*, pages 135–140. IEEE, 2014.

[23] Larocca Conte Gabriele; Aleksinski Adam; Liao Ashley. Data from: Eocene shark teeth from peninsular antarctica: Windows to habitat use and paleoceanography [data set], 2024. Accessed: 2025-04-10.

[24] Assmann Cora ; Gerlach Roman ; Lang Kevin ; Neute Nadine ; Rex Jessica. Fair-checker: supporting digital resource findability and reuse with knowledge graphs and semantic web standards. *Journal of Biomedical Semantics*, 14, 04 2023.

[25] Australian Research Data Commons. Ardc-fair-tool. `https://ardc.edu.au/resource/fair-data-self-assessment-tool/`. Accessed: 2024-10-24.

[26] Miriam Baglioni, Paolo Manghi, Andrea Mannocci, and Alessia Bardi. We can make a better use of orcid: five observed misapplications. *Data Science Journal*, 20(1), 2021.

[27] Tian Bai, Brian L Egleston, Richard Bleicher, and Slobodan Vucetic. Medical concept representation learning from multi-source data. In *IJCAI: Proceedings of the Conference*, volume 2019, page 4897. NIH Public Access, 2019.

[28] Tian Bai and Slobodan Vucetic. Improving medical code prediction from clinical text via incorporating online knowledge sources. In *The World Wide Web Conference*, pages 72–82, 2019.

[29] Vidhya Balasubramanian, Sooryanarayan Gobu Doraisamy, and Navaneeth Kumar Kanakarajan. A multimodal approach for extracting content descriptive metadata from lecture videos. *Journal of Intelligent Information Systems*, 46:121–145, 2016.

[30] Vidhya Balasubramanian, Sooryanarayan Gobu Doraisamy, and Navaneeth Kumar Kanakarajan. A multimodal approach for extracting content descriptive metadata from lecture videos. *J. Intell. Inf. Syst.*, 46(1):121–145, 2016.

[31] Weidong Bao, Hongfei Lin, Yijia Zhang, Jian Wang, and Shaowu Zhang. Medical code prediction via capsule networks and icd knowledge. *BMC Medical Informatics and Decision Making*, 21(2):1–12, 2021.

[32] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. Multi-label classification of patient notes: case study on icd code assignment. In *Workshops at the thirty-second AAAI conference on artificial intelligence*, 2018.

[33] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620, 2017.

[34] Fredrik Bergholm. Edge focusing. *IEEE transactions on pattern analysis and machine intelligence*, (6):726–741, 1987.

[35] Donna Bergmark. Automatic extraction of reference linking information from online documents. Technical report, Cornell University, 2000.

[36] G Bharathi Mohan, R Prasanna Kumar, P Vishal Krishh, A Keerthinathan, G Lavanya, Meka Kavya Uma Meghana, Sheba Sulthana, and Srinath Doss. An analysis of large language models: their impact and potential applications. *Knowledge and Information Systems*, 66(9):5047–5070, 2024.

[37] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR, 23–29 Jul 2023.

[38] Bioinformatique France. Fairchecker. `https://fair-checker.france-bioinformatique.fr/`. Accessed: 2025-03-10.

[39] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[40] Biplob Biswas, Thai-Hoang Pham, and Ping Zhang. Transicd: Transformer based code-wise attention model for explainable icd coding. In Allan Tucker, Pedro Henriques Abreu, Jaime Cardoso, Pedro Pereira Rodrigues, and David Riaño, editors, *Artificial Intelligence in Medicine*, pages 469–478, Cham, 2021. Springer International Publishing.

[41] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[42] Arnim Bleier, Kenan Erdogan, Christian Kahmann, and Lisa Posch. Gesis notebooks: Online reproducible computational analysis for the social sciences, 2022.

[43] Kurt D Bollacker, Steve Lawrence, and C Lee Giles. Citeseer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *Proceedings of the second international conference on Autonomous agents*, pages 116–123. ACM, 1998.

[44] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

[45] Alex Bottle and Paul Aylin. Intelligent information: a national system for monitoring clinical performance. *Health services research*, 43(1p1):10–31, 2008.

[46] Zeyd Boukhers, Shriharsh Ambhore, and Steffen Staab. An end-to-end approach for extracting and segmenting high-variance references from pdf documents. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 186–195, 2019.

[47] Zeyd Boukhers and Nagaraj Bahubali Asundi. Whois? deep author name disambiguation using bibliographic data. In *Linking Theory and Practice of Digital Libraries: 26th International Conference on Theory and Practice of Digital Libraries, TPDL 2022, Padua, Italy, September 20–23, 2022, Proceedings*, pages 201–215. Springer, 2022.

[48] Zeyd Boukhers and Nagaraj Bahubali Asundi. Deep author name disambiguation using dblp data. *International Journal on Digital Libraries*, pages 1–11, 2023.

[49] Zeyd Boukhers, Nada Beili, Timo Hartmann, Prantik Goswami, and Muhammad Arslan Zafar. Mexpub: Deep transfer learning for metadata extraction from german publications. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2021.

[50] Zeyd Boukhers, Nada Beili, Timo Hartmann, Prantik Goswami, and Muhammad Arslan Zafar. Mexpub: Deep transfer learning for metadata extraction from german publications. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 250–253. IEEE, 2021.

[51] Zeyd Boukhers, Arnim Bleier, Yeliz Ucer Yediel, Mio Hienstorfer-Heitmann, Mehrshad Jaberansary, Sascha Welten, Adamantios Koumpis, and Oya Beyan. Padme-sosci: A platform for analytics and distributed machine learning for the social sciences. In *2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 251–252. IEEE, 2023.

[52] Zeyd Boukhers and Azeddine Bouabdallah. Vision and natural language for metadata extraction from scientific pdf documents: a multimodal approach. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, pages 1–5, 2022.

[53] Zeyd Boukhers, Azeddine Bouabdallah, Cong Yang, and Jan Jürjens. Beyond trading data: The hidden influence of public awareness and interest on cryptocurrency volatility. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 142–151, 2023.

[54] Zeyd Boukhers, Prantik Goswami, and Jan Jürjens. Knowledge guided multi-filter residual convolutional neural network for icd coding from clinical text. *Neural Computing and Applications*, 35(24):17633–17644, 2023.

[55] Zeyd Boukhers, Timo Hartmann, and Jan Jürjens. Coin: Counterfactual image generation for visual question answering interpretation. *Sensors*, 22(6):2245, 2022.

[56] Zeyd Boukhers, AmeerAli Khan, Qusai Ramadan, and Cong Yang. Large language model in medical informatics: Direct classification and enhanced text representations for automatic icd coding. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2024. Accepted for publication.

[57] Zeyd Boukhers and Cong Yang. Comparison of feature learning methods for metadata extraction from pdf scientific documents. *IEEE Transactions on Knowledge and Data Engineering*, 2024. Under review.

[58] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[59] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss,

Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[60] Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4, 2024.

[61] Leonardo Candela, Dario Mangione, and Gina Pavone. The fair assessment conundrum: Reflections on tools and metrics. *Data Science Journal*, May 2024.

[62] Kris Cao and Marek Rei. A joint model for word embedding and word morphology. *arXiv preprint arXiv:1606.02601*, 2016.

[63] J Harry Caufield et al. A reference set of curated biomedical data and metadata from clinical case reports. *Scientific data*, 2018.

[64] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. Interpretability of deep learning models: A survey of results. In *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*, pages 1–6. IEEE, 2017.

[65] Nathanael Chambers and Dan Jurafsky. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 976–986. Association for Computational Linguistics, 2011.

[66] Ashis Kumar Chanda, Tian Bai, Ziyu Yang, and Slobodan Vucetic. Improving medical term embeddings using umls metathesaurus. *BMC Medical Informatics and Decision Making*, 22(1):1–12, 2022.

[67] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[68] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3), mar 2024.

[69] Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. *arXiv preprint arXiv:1807.03121*, 2018.

[70] Chien-Chih Chen, Kai-Hsiang Yang, Chuen-Liang Chen, and Jan-Ming Ho. Bibpro: A citation parser based on sequence alignment. *IEEE Transactions on Knowledge and Data Engineering 24, 2 (2012)*, page 236–250, 2012.

[71] Chien-Chih Chen, Kai-Hsiang Yang, Hung-Yu Kao, and Jan-Ming Ho. Bibpro: A citation parser based on sequence alignment techniques. In *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1175–1180. IEEE, 2008.

[72] Irene Y Chen, Emily Alsentzer, Hyesun Park, Richard Thomas, Babina Gosangi, Rahul Gujrathi, and Bharti Khurana. Intimate partner violence and injury prediction from radiology reports. In *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*, 2020.

[73] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10800–10809, 2020.

[74] Julien Chevallier, Dominique Guégan, and Stéphane Goutte. Is it possible to forecast the price of bitcoin? *Forecasting*, 3(2):377–420, 2021.

[75] Jason P. C. Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *CoRR*, abs/1511.08308, 2015.

[76] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[77] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare

representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 787–795, 2017.

[78] David LEE Kuo Chuen, Li Guo, and Yu Wang. Cryptocurrency: A new investment opportunity? *The Journal of Alternative Investments*, 20(3):16–40, 2017.

[79] Daniel J. B. Clarke, Lily Wang, Alex Jones, Megan L. Wojciechowicz, Denis Torre, Kathleen M. Jagodnik, Sherry L. Jenkins, Peter McQuilton, Zachary Flamholz, Moshe C. Silverstein, Brian M. Schilder, Kimberly Robasky, Claris Castillo, Ray Idaszak, Stanley C. Ahalt, Jason Williams, Stephan Schurer, Daniel J. Cooper, Ricardo de Miranda Azevedo, Juergen A. Klenk, Melissa A. Haendel, Jared Nedzel, Paul Avillach, Mary E. Shimoyama, Rayna M. Harris, Meredith Gamble, Rudy Poten, Amanda L. Charbonneau, Jennie Larkin, C. Titus Brown, Vivien R. Bonazzi, Michel J. Dumontier, Susanna-Assunta Sansone, and Avi Ma'ayan. Fairshake: toolkit to evaluate the findability, accessibility, interoperability, and reusability of research digital resources. *bioRxiv*, 2019.

[80] Isaac G Councill, C Lee Giles, and Min-Yen Kan. Parscit: an open-source crf reference string parsing package. In *LREC*, volume 8, pages 661–667, 2008.

[81] Isaac G Councill, C Lee Giles, and Min-Yen Kan. Parscit: an open-source crf reference string parsing package. *LREC, Vol. 8.*, page 661–667, 2008.

[82] Zhi Da, Joseph Engelberg, and Pengjie Gao. In search of attention. *The journal of finance*, 66(5):1461–1499, 2011.

[83] Suyang Dai, Yuxia Ding, Zihan Zhang, Wenxuan Zuo, Xiaodi Huang, and Shanfeng Zhu. Grantextractor: Accurate grant support information extraction from biomedical fulltext based on bi-lstm-crf. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(1):205–215, 2019.

[84] Christoph Dann, Yishay Mansour, and Mehryar Mohri. Reinforcement learning can be more efficient with multiple rewards. In *International Conference on Machine Learning*, pages 6948–6967. PMLR, 2023.

[85] Marilena Daquino, Silvio Peroni, David Shotton, Giovanni Colavizza, Behnam Ghavimi, Anne Lauscher, Philipp Mayr, Matteo Romanello, and Philipp Zumstein. The opencitations data model. In *International semantic web conference*, pages 447–463. Springer, 2020.

[86] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100, 2017.

[87] Joyeeta Datta, Niclas Doll, Qusai Ramadan, and Zeyd Boukhers. Exploring the limits of model compression in llms: A knowledge distillation study on qa tasks. In *2025 Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2025.

[88] Min-Yuh Day, Richard Tzong-Han Tsai, Cheng-Lung Sung, Chiu-Chen Hsieh, Cheng-Wei Lee, Shih-Hung Wu, Kun-Pin Wu, Chorng-Shyong Ong, and Wen-Lian Hsu. Reference metadata extraction using a hierarchical knowledge representation framework. *Decision Support Systems*, 43(1):152–167, 2007.

[89] Min-Yuh Day, Richard Tzong-Han Tsai, Cheng-Lung Sung, Chiu-Chen Hsieh, Cheng-Wei Lee, Shih-Hung Wu, Kun-Pin Wu, Chorng-Shyong Ong, and Wen-Lian Hsu. Reference metadata extraction using a hierarchical knowledge representation framework. *Decision Support Systems*, 43(1):152–167, 2007.

[90] Luciano RS de Lima, Alberto HF Laender, and Berthier A Ribeiro-Neto. A hierarchical approach to the automatic categorization of medical documents. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 132–139, 1998.

[91] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[92] Emily Denton, Ben Hutchinson, Margaret Mitchell, Timnit Gebru, and Andrew Zaldivar. Image counterfactual sensitivity analysis for detecting unintended bias. *arXiv preprint arXiv:1906.06439*, 2019.

[93] Anusuriya Devaraju, Robert Huber, Mustapha Mokrane, Patricia Herterich, L. Cepinskas, Jerry de Vries, Herve L'Hours, Joy Davidson, and Angus White. *FAIRsFAIR Data Object Assessment Metrics.* ", October 2020.

[94] Devaraju, Anusuriya and Mokrane, Mustapha and Cepinskas, Linas and Huber, Robert and Herterich, Patricia and Vries, Jerry and Akerman, Vesa and L'Hours, Hervé and Davidson, Joy and Diepenbroek, Michael. From Conceptualization to Implementation: FAIR Assessment of Research Data Objects. *Data Science Journal*, 20, 02 2021.

[95] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[96] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[97] Xiaolin Diao, Yanni Huo, Shuai Zhao, Jing Yuan, Meng Cui, Yuxin Wang, Xiaodan Lian, and Wei Zhao. Automated icd coding for primary diagnosis via clinically interpretable machine learning. *International Journal of Medical Informatics*, 153:104543, 2021.

[98] Ying Ding, Gobinda Chowdhury, Schubert Foo, et al. Template mining for the extraction of citation from digital documents. In *Proceedings of the Second Asian Digital Library Conference, Taiwan*, pages 47–62, 1999.

[99] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[100] Yichao Du, Pengfei Luo, Xudong Hong, Tong Xu, Zhe Zhang, Chao Ren, Yi Zheng, and Enhong Chen. Inheritance-guided hierarchical assignment for clinical automatic diagnosis. In *International Conference on Database Systems for Advanced Applications*, pages 461–477. Springer, 2021.

[101] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

[102] Aniruddha Dutta, Saket Kumar, and Meheli Basu. A gated recurrent unit approach to bitcoin price prediction. *Journal of Risk and Financial Management*, 13(2):23, 2020.

[103] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment*, 11(11):1454–1467, 2018.

[104] European Commission. Commission Recommendation (EU) 2019/243 on a European Electronic Health Record exchange format, 2019.

[105] European Organization For Nuclear Research and OpenAIRE. Zenodo.org, 2013. Accessed: 2025-04-10.

[106] European Parliament. Artificial Intelligence Act: AI-Act, 13 June 2024.

[107] European Parliament. Proposal for a REGULATION OF THE EP AND OF THE COUNCIL on the European Health Data Space: COM/2022/197 final, 3.5.2022.

[108] FAIRsFAIR Group. F-UJI. `https://www.fairsfair.eu/f-uji-automated-fair-data-assessment-tool`. Accessed: 2025-03-10.

[109] Xiaoming Fan, Jianyong Wang, Xu Pu, Lizhu Zhou, and Bing Lv. On graph-based name disambiguation. *Journal of Data and Information Quality (JDIQ)*, 2(2):1–23, 2011.

[110] Amal Feriani, Di Wu, Yi Tian Xu, Jimmy Li, Seowoo Jang, Ekram Hossain, Xue Liu, and Gregory Dudek. Multiobjective load balancing for multiband downlink cellular networks: A meta-reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 40(9):2614–2629, 2022.

[111] Carlos Fernández-Loría, Foster Provost, and Xintian Han. Explaining data-driven decisions made by ai systems: the counterfactual approach. *arXiv preprint arXiv:2001.07417*, 2020.

[112] José Carlos Ferrão, Filipe Janela, Mónica Duarte Oliveira, and Henrique MG Martins. Using structured ehr data and svm to support icd-9-cm coding. In *2013 ieee international conference on healthcare informatics*, pages 511–516. IEEE, 2013.

[113] Anderson A Ferreira, Marcos André Gonçalves, and Alberto HF Laender. A brief survey of automatic methods for author name disambiguation. *Acm Sigmod Record*, 41(2):15–26, 2012.

[114] Anderson A Ferreira, Marcos André Gonçalves, and Alberto HF Laender. *Automatic disambiguation of author names in bibliographic repositories*. Morgan & Claypool Publishers, 2020.

[115] Anderson A Ferreira, Adriano Veloso, Marcos André Gonçalves, and Alberto HF Laender. Effective self-training author name disambiguation in scholarly digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 39–48, 2010.

[116] Jeremy Foxcroft, Adrian d'Alessandro, and Luiza Antonie. Name2vec: Personal names embeddings. In *Canadian Conference on Artificial Intelligence*, pages 505–510. Springer, 2019.

[117] Jason Alan Fries. Brundlefly at semeval-2016 task 12: Recurrent neural networks vs. joint inference for clinical temporal information extraction. *arXiv preprint arXiv:1606.01433*, 2016.

[118] Tamiru Gabusho, Chala Gelana, and Ismael Hussein. Effect of financial inclusion on rural-urban households' economic welfare west hararghe zone, evidence's from some selected woredaa [data set], 2025. Accessed: 2025-04-10.

[119] Alban Gaignard, Thomas Rosnet, Frédéric Lamotte, Vincent Lefort, and Marie-Dominique Devignes. Fair assessment tools: An evaluation of assessment tools of data sets according to the fair principles. *Journal of Biomedical Semantics*, 07 2023.

[120] Kavita Ganesan. ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks. *arXiv preprint arXiv:1803.01937*, 2018.

[121] Walter Gautschi. *Numerical analysis*. Springer Science & Business Media, 2011.

[122] Aryo Pradipta Gema, Pasquale Minervini, Luke Daines, Tom Hope, and Beatrice Alex. Parameter-Efficient Fine-Tuning of LLaMA for the Clinical Domain.

[123] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015.

[124] Behnam Ghavimi, Wolfgang Otto, and Philipp Mayr. Exmatcher: Combining features based on reference strings and segments to enhance citation matching, 2019. Manuscript submitted for publication.

[125] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[126] Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. Vice: visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 531–535, 2020.

[127] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[128] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[129] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[130] Aaron Grattafiori et al. The Llama 3 Herd of Models.

[131] Trond Grenager, Dan Klein, and Christopher D. Manning. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 371–378, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[132] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[133] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

[134] Paul Guillot, Martin Bøgsted, and Charles Vesteghem. Fair sharing of health data: a systematic review of applicable solutions. *Health and Technology*, 13, 11 2023.

[135] Aynur Guluzade et al. Appendix - ELMTEX.

[136] Aynur Guluzade et al. ELMTEX Dataset.

[137] Aynur Guluzade, Naguib Heiba, Zeyd Boukhers, Florim Hamiti, Jahid Hasan Polash, Yehya Mohamad, and Carlos A Velasco. Elmtex: Fine-tuning llms for structured clinical information extraction. a case study on clinical reports. In *International Conference on Artificial Intelligence in Medicine*, pages 181–185. Springer, 2025.

[138] Akshay Kumar Gupta. Survey of visual question answering: Datasets and techniques. *arXiv preprint arXiv:1705.03865*, 2017.

[139] Udo Hahn and Michel Oleynik. Medical information extraction in the age of deep learning. *Yearbook of medical informatics*, 2020.

[140] Hui Han, C.L. Giles, E. Manavoglu, Hongyuan Zha, Zhenyue Zhang, and E.A. Fox. Automatic document metadata extraction using support vector machines. In *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, pages 37–48, 2003.

[141] Hui Han, Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004.*, pages 296–305. IEEE, 2004.

[142] Hui Han, Eren Manavoglu, Hongyuan Zha, Kostas Tsioutsiouliklis, C. Lee Giles, and Xiangmin Zhang. Rule-based word clustering for document metadata extraction. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, page 1049–1053, New York, NY, USA, 2005. Association for Computing Machinery.

[143] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774, 2011.

[144] James V Hansen, James B McDonald, and Ray D Nelson. Time series prediction with genetic-algorithm designed neural networks: An empirical comparison with modern statistical models. *Computational Intelligence*, 15(3):171–184, 1999.

[145] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

[146] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[147] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[148] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[149] Ginny Hendricks, Dominika Tkaczyk, Jennifer Lin, and Patricia Feeney. Crossref: The sustainable source of community-owned scholarly metadata. *Quantitative Science Studies*, 1(1):414–427, 2020.

[150] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language. *arXiv preprint arXiv:1806.09809*, 2018.

[151] Linus Hermansson, Tommi Kerola, Fredrik Johansson, Vinay Jethava, and Devdatt Dubhashi. Entity disambiguation in anonymized graphs using graph kernels. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1037–1046, 2013.

[152] Erik Hetzner. A simple method for citation metadata extraction using hidden markov models. In *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '08, pages 280–284, New York, NY, USA, 2008. ACM.

[153] Daniel Hienert, Frank Sawitzki, and Philipp Mayr. Digital library research in action - supporting information retrieval in sowiport. *D-Lib Magazine*, 21(3/4), 2015.

[154] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.

[155] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[156] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, 2011.

[157] Mohammad Asiful Hossain, Rezaul Karim, Ruppa Thulasiram, Neil DB Bruce, and Yang Wang. Hybrid deep learning model for stock price prediction. In *2018 ieee symposium series on computational intelligence (ssci)*, pages 1837–1844. IEEE, 2018.

[158] Azam Hosseini, Behnam Ghavimi, Dagmar Kern, and Philipp Mayr. Excite–a toolchain to extract, match and publish open literature references. In *ACM/IEEE-CS joint conference on Digital libraries*, pages 432–433, 2019.

[159] Oumaima Hourrane, Sara Mifrah, Nadia Bouhriz, Mohamed Rachdi, et al. Using deep learning word embeddings for citations similarity in academic papers. In *International Conference on Big Data, Cloud and Applications*, pages 185–196. Springer, 2018.

[160] Kristen Howell, Gwen Christian, Pavel Fomitchov, Gitit Kehat, Julianne Marzulla, Leanne Rolston, Jadin Tredup, Ilana Zimmerman, Ethan Selfridge, and Joseph Bradley. The economic trade-offs of large language models: A case study. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 248–267. Association for Computational Linguistics, 2023.

[161] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[162] Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, et al. Improving large language models for clinical named entity recognition via prompt engineering. *Journal of the American Medical Informatics Association*, 2024.

[163] Xin Huang, Wenbin Zhang, Yiyi Huang, Xuejiao Tang, Mingli Zhang, Jayachander Surbiryala, Vasileios Iosifidis, Zhen Liu, and Ji Zhang. Lstm based sentiment analysis for cryptocurrency prediction. *arXiv preprint arXiv:2103.14804*, 2021.

[164] Yukun Huang, Yanda Chen, Zhou Yu, and Kathleen McKeown. In-context Learning Distillation: Transferring Few-shot Learning Ability of Pre-trained Language Models. *arXiv preprint arXiv:2212.10670*, 2022.

[165] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[166] E. A. Huerta, Ben Blaiszik, L. Catherine Brinson, Kristofer E. Bouchard, Daniel Diaz, Caterina Doglioni, Javier M. Duarte, Murali Emani, Ian Foster, Geoffrey Fox, Philip Harris, Lukas Heinrich, Shantenu Jha, Daniel S. Katz, Volodymyr Kindratenko, Christine R. Kirkpatrick, Kati Lassila-Perini, Ravi K. Madduri, Mark S. Neubauer, Fotis E. Psomopoulos, Avik Roy, Oliver Rübel, Zhizhen Zhao, and Ruike Zhu. Fair for ai: An interdisciplinary and international community building perspective. *Scientific Data*, 10(1), July 2023.

[167] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report, 2024.

[168] Ijaz Hussain and Sohail Asghar. A survey of author name disambiguation techniques: 2010-2016. *Knowledge Eng. Review*, 32:e22, 2017.

[169] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.

[170] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[171] J. Scott MARCUS, Bertin MARTENS, Anne BUCHER Christophe CARUGATI, and and Ilsa GODLOVITCH. The European Health Data Space.

[172] Huisu Jang and Jaewook Lee. An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *Ieee Access*, 6:5427–5437, 2017.

[173] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th international conference on knowledge capture*, pages 243–246, 2019.

[174] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[175] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[176] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online, November 2020. Association for Computational Linguistics.

[177] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[178] Sarah Jones and Marjan Grootveld. How fair are your data?, July 2021.

[179] Hee-Sung Jung. Ice/water classification maps from study entitled "an arctic sea ice concentration data record on a 6.25 km polar stereographic grid from 3 years of landsat-8 imagery" [data set], 2025. Accessed: 2025-04-10.

[180] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and Applications of Large Language Models. *arXiv preprint arXiv:2307.10169*, 2023.

[181] Asanee Kawtrakul and Chaiyakorn Yingsaeree. A unified framework for automatic metadata extraction from electronic document. In *Proceedings of The International Advanced Digital Library Conference. Nagoya, Japan*, 2005.

[182] Madian Khabsa, Pucktada Treeratpituk, and C Lee Giles. Large scale author name disambiguation in digital libraries. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 41–42. IEEE, 2014.

[183] Madian Khabsa, Pucktada Treeratpituk, and C Lee Giles. Online person name disambiguation with constraints. In *Proceedings of the 15th acm/ieee-cs joint conference on digital libraries*, pages 37–46, 2015.

[184] Kaustubh Khare, Omkar Darekar, Prafull Gupta, and VZ Attar. Short term stock price prediction using deep learning. In *2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*, pages 482–486. IEEE, 2017.

[185] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907. Association for Computational Linguistics, 2020.

[186] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

[187] Kunho Kim, Athar Sefid, and C Lee Giles. Learning cnf blocking for large-scale author name disambiguation. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 72–80, 2020.

[188] Kunho Kim, Athar Sefid, Bruce A Weinberg, and C Lee Giles. A web service for author name disambiguation in scholarly databases. In *2018 IEEE International Conference on Web Services (ICWS)*, pages 265–273. IEEE, 2018.

[189] Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, and Chang Hun Kim. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PloS one*, 11(8):e0161197, 2016.

[190] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DISTILLM: towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 24872–24895, 2024.

[191] Sergei Koltcov, Vera Ignatenko, Zeyd Boukhers, and Steffen Staab. Analyzing the influence of hyper-parameters and regularizers of topic modeling in terms of renyi entropy. *Entropy*, 22(4):394, 2020.

[192] Lingxiao Kong, Cong Yang, Susanne Neufang, Oya Deniz Beyan, and Zeyd Boukhers. Emorl: Ensemble multi-objective reinforcement learning for efficient and flexible llm fine-tuning. In *2025 Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2025.

[193] Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. Automatic icd-10 classification of cancers from free-text death certificates. *International journal of medical informatics*, 84(11):956–965, 2015.

[194] Martin Körner, Behnam Ghavimi, Philipp Mayr, Heinrich Hartmann, and Steffen Staab. Evaluating reference string extraction using line-based conditional random fields: A case study with german language publications. In *European Conference on Advances in Databases and Information Systems*, pages 137–145. Springer, 2017.

[195] N.A. Krans, A. Ammar, P. Nymark, E.L. Willighagen, M.I. Bakker, and J.T.K. Quik. Fair assessment tools: evaluating use and performance. *NanoImpact*, 27:100402, 2022.

[196] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

[197] Ladislav Kristoufek. What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. *PloS one*, 10(4):e0123923, 2015.

[198] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

[199] Deepak Kumar and SK Rath. Predicting the trends of price for ethereum using deep learning techniques. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pages 103–114. Springer, 2020.

[200] Pranjal Kumar. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260, 2024.

[201] Martin Körner, Behnam Ghavimi, Philipp Mayr, Heinrich Hartmann, and Steffen Staab. Evaluating reference string extraction using line-based conditional random fields: A case study with german language publications. In *New Trends in Databases and Information Systems*, volume 767, pages 137–145. Springer International Publishing, 2017.

[202] Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A Hearst. Keep it simple: Unsupervised simplification of multi-paragraph text. *arXiv preprint arXiv:2107.03444*, 2021.

[203] John Lafferty, Andrew McCallum, Fernando Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA, 2001.

[204] Connor Lamon, Eric Nielsen, and Eric Redondo. Cryptocurrency price prediction using news and social media sentiment. *SMU Data Sci. Rev*, 1(3):1–22, 2017.

[205] Mohamed Yassine Landolsi, Lobna Hlaoua, and Lotfi Ben Romdhane. Information extraction from electronic medical documents: state of the art and future research directions. *Knowledge and Information Systems*, 2023.

[206] Leah S Larkey and W Bruce Croft. Combining classifiers in text categorization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 289–297, 1996.

[207] June-Goo Lee, Sanghoon Jun, Young-Won Cho, Hyunna Lee, Guk Bae Kim, Joon Beom Seo, and Namkug Kim. Deep learning in medical imaging: general overview. *Korean journal of radiology*, 18(4):570–584, 2017.

[208] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287*, 2019.

[209] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 7871. Association for Computational Linguistics, 2020.

[210] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330. Association for Computational Linguistics, 2020.

[211] Chang Li, Dongjin Song, and Dacheng Tao. Multi-task recurrent neural networks and higher-order markov random fields for stock price movement prediction: Multi-task rnn and higer-order mrfs for stock price classification. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1141–1151, 2019.

[212] Fei Li and Hong Yu. Icd coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 05, pages 8180–8187, 2020.

[213] Guoliang Li, Xuanhe Zhou, and Xinyang Zhao. Llm for data management. *Proc. VLDB Endow.*, 17(12):4213–4216, August 2024.

[214] Huajing Li, Isaac Councill, Wang-Chien Lee, and C Lee Giles. Citeseerx: an architecture and web service design for an academic document search engine. In *Proceedings of the 15th international conference on World Wide Web*, pages 883–884, 2006.

[215] Qing Li, Jianlong Fu, Dongfei Yu, Tao Mei, and Jiebo Luo. Tell-and-answer: Towards explainable visual question answering using attributes and captions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1338–1346, 2018.

[216] Baihan Lin. Reinforcement learning in large language models (llms): The rise of ai language giants. In *Reinforcement Learning Methods in Speech and Language Technology*, pages 147–156. Springer, 2024.

[217] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[218] Zinan Lin, Vyas Sekar, and Giulia Fanti. Why spectral normalization stabilizes gans: Analysis and improvements. *arXiv preprint arXiv:2009.02773*, 2020.

[219] Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. Hierarchical label-wise attention transformer model for explainable icd coding. *Journal of Biomedical Informatics*, 133:104161, September 2022.

[220] Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. Automated icd coding using extreme multi-label long text transformer-based models. *Artificial Intelligence in Medicine*, 144:102662, 2023.

[221] Runtao Liu, Liangcai Gao, Dong An, Zhuoren Jiang, and Zhi Tang. Automatic document metadata extraction based on deep networks. In Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng, and Yu Hong, editors, *Natural Language Processing and Chinese Computing*, pages 305–317, Cham, 2018. Springer International Publishing.

[222] Wanli Liu, Rezarta Islamaj Doğan, Sun Kim, Donald C Comeau, Won Kim, Lana Yeganova, Zhiyong Lu, and W John Wilbur. Author name disambiguation for p ub m ed. *Journal of the Association for Information Science and Technology*, 65(4):765–781, 2014.

[223] Yue Liu, Tao Ge, Kusum S Mathews, Heng Ji, and Deborah L McGuinness. Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion. *arXiv preprint arXiv:1804.04225*, 2018.

[224] Ioannis E Livieris, Emmanuel Pintelas, Stavros Stavroyiannis, and Panagiotis Pintelas. Ensemble deep learning models for forecasting cryptocurrency time-series. *Algorithms*, 13(5):121, 2020.

[225] Llama AI Team @ Meta. The llama 3 herd of models, 2024.

[226] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld. S2orc: The semantic scholar open research corpus. *arXiv preprint arXiv:1911.02782*, 2019.

[227] Patrice Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In Maristella Agosti, José Borbinha, Sarantos Kapidakis, Christos Papatheodorou, and Giannis Tsakonas, editors, *Research and Advanced Technology for Digital Libraries*, pages 473–474, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[228] Gilles Louppe, Hussein T Al-Natsheh, Mateusz Susik, and Eamonn James Maguire. Ethnicity sensitive author disambiguation using semi-supervised learning. In *international conference on knowledge engineering and the semantic web*, pages 272–287. Springer, 2016.

[229] Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology*, 9(7):237–248, 2016.

[230] Fenglong Ma, Quanzeng You, Houping Xiao, Radha Chitta, Jing Zhou, and Jing Gao. Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 743–752, 2018.

[231] Isaac Madan, Shaurya Saluja, and Aojia Zhao. Automated bitcoin trading via machine learning algorithms. *URL: http://cs229. stanford. edu/proj2014/Isaac% 20Madan*, 20, 2015.

[232] Shanza Zafar Malik, Khalid Iqbal, Muhammad Sharif, Yaser Ali Shah, Amaad Khalil, M Abeer Irfan, and Joanna Rosak-Szyrocka. Attention-aware with stacked embedding for sentiment analysis of student feedback through deep learning techniques. *PeerJ Computer Science*, 10:e2283, 2024.

[233] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. *Advances in neural information processing systems*, 27:1682–1690, 2014.

[234] Paolo Manghi, Claudio Atzori, Alessia Bardi, Miriam Baglioni, Jochen Schirrwagen, Harry Dimitropoulos, Sandro La Bruzzo, Ioannis Foufoulas, Andrea Mannocci, Marek Horst, et al. Openaire research graph dump. 2022.

[235] Kevin McGeechan, Petra Macaskill, Les Irwig, and Patrick MM Bossuyt. An assessment of the relationship between clinical utility and predictive ability measures and the impact of mean risk in the population. *BMC medical research methodology*, 14(1):1–12, 2014.

[236] Sean McNally, Jason Roche, and Simon Caton. Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*, pages 339–343. IEEE, 2018.

[237] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[238] Do June Min, Verónica Pérez-Rosas, Kenneth Resnicow, and Rada Mihalcea. PAIR: Prompt-aware margIn ranking for counselor reflection scoring in motivational interviewing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 148–158, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[239] Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3d environments with visual goal prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2667–2678, 2018.

[240] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[241] Henk F. Moed. *Citation Analysis in Research Evaluation*. Information Science & Knowledge Management. Springer-Verlag, Berlin, Heidelberg, 2005.

[242] Aditya Mohan, Carolin Benjamins, Konrad Wienecke, Alexander Dockhorn, and Marius Lindauer. Autorl hyperparameter landscapes. *arXiv preprint arXiv:2304.02396*, 2023.

[243] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[244] Milad Moradi, Kathrin Blagec, Florian Haberl, and Matthias Samwald. GPT-3 Models are Poor Few-Shot Learners in the Biomedical Domain.

[245] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(1):18–33, 2020.

[246] Mohammed Mudassir, Shada Bennbaia, Devrim Unal, and Mohammad Hammoudeh. Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*, pages 1–15, 2020.

[247] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.

[248] Mark-Christoph Müller. Semantic author name disambiguation with word embeddings. In *International Conference on Theory and Practice of Digital Libraries*, pages 300–311. Springer, 2017.

[249] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[250] Agnieszka Mykowiecka, Małgorzata Marciniak, and Anna Kupść. Rule-based information extraction from patients' clinical data. *Journal of biomedical informatics*, 2009.

[251] Shyamala G Nadathur. Maximising the value of hospital administrative datasets. *Australian Health Review*, 34(2):216–223, 2010.

[252] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 42–51, 2018.

[253] Sergey Nasekin and Cathy Yi-Hsuan Chen. Deep learning-based cryptocurrency sentiment construction. *Available at SSRN 3310784*, 2019.

[254] P Raghavendra Nayaka and Rajeev Ranjan. An efficient framework for metadata extraction over scholarly documents using ensemble cnn and bilstm technique. In *2023 2nd International Conference for Innovation in Technology (INOCON)*, pages 1–9. IEEE, 2023.

[255] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: a convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30, 2016.

[256] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. {*Deepr*}: a convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30, 2016.

[257] Thien Hai Nguyen and Kiyoaki Shirai. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of*

the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1354–1364, 2015.

[258] Paul Nickerson, Patrick Tighe, Benjamin Shickel, and Parisa Rashidi. Deep neural network architectures for forecasting analgesic response. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2966–2969. IEEE, 2016.

[259] Mahla Nikou, Gholamreza Mansourfar, and Jamshid Bagherzadeh. Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, 26(4):164–174, 2019.

[260] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12700–12710, 2021.

[261] Caio Nóbrega and Leandro Marinho. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1671–1678, 2019.

[262] MM Nugues and CM Roberts. Coral mortality and interaction with algae in relation to sedimentation. *Coral reefs*, 22(4):507–516, 2003.

[263] Siddhartha Nuthakki, Sunil Neela, Judy W. Gichoya, and Saptarshi Purkayastha. Natural language processing of MIMIC-III clinical notes for identifying diagnosis and procedures with neural networks.

[264] Yuki Okano, Kotaro Funakoshi, Ryo Nagata, and Manabu Okumura. Generating dialog responses with specified grammatical items for second language learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 184–194, 2023.

[265] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[266] Emma O'neil, João Sedoc, Diyi Yang, Haiyi Zhu, and Lyle Ungar. Automatic reflection generation for peer-to-peer counseling. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 62–75, 2023.

[267] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Med-HALT: Medical domain hallucination test for large language models. In Jing Jiang, David Reitter, and Shumin Deng, editors, *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 314–334, 2023.

[268] Jingjing Pan, Yash Goyal, and Stefan Lee. Question-conditioned counterfactual image generation for vqa. *arXiv preprint arXiv:1911.06352*, 2019.

[269] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[270] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*, 2024.

[271] Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems 15 - Proceedings of the 2002 Conference, NIPS 2002*. Neural information processing systems foundation, 1 2003.

[272] Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 3–3, 2018.

[273] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models, 2023.

[274] Cheng Peng, XI Yang, Kaleb E Smith, Zehao Yu, Aokun Chen, Jiang Bian, and Yonghui Wu. Model tuning or prompt tuning? a study of large language models for clinical concept and relation extraction. *Journal of biomedical informatics*, 2024.

[275] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42(4):963–979, July 2006.

[276] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42(4):963–979, 2006.

[277] Xueping Peng, Guodong Long, Tao Shen, Sen Wang, Zhendong Niu, and Chengqi Zhang. Mimo: mutual integration of patient journey and medical ontology for healthcare representation learning. *arXiv preprint arXiv:2107.09288*, 2021.

[278] Verónica Pérez-Rosas, Ken Resnicow, Rada Mihalcea, et al. Dynamic reward adjustment in multi-reward reinforcement learning for counselor reflection generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5437–5449, 2024.

[279] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237, 2014.

[280] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[281] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language Models as Knowledge Bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics.

[282] Luiza Petrosyan, Rafael Aleixandre-Benavent, Fernanda Peset, Juan Carlos Valderrama-Zurián, Antonia Ferrer-Sapena, and Andrea Sixto-Costoya. Fair degree assessment in agriculture datasets using the f-uji tool. *Ecological Informatics*, 76:102126, 2023.

[283] Emmanuel Pintelas, Ioannis E Livieris, Stavros Stavroyiannis, Theodore Kotsilieris, and Panagiotis Pintelas. Investigating the problem of cryptocurrency price prediction: a deep learning approach. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 99–110. Springer, 2020.

[284] Aaditya Prakash, Siyuan Zhao, Sadid A Hasan, Vivek Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Condensed memory networks for clinical diagnostic inferencing. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[285] Animesh Prasad, Manpreet Kaur, and Min-Yen Kan. Neural parscit: a deep learning-based reference string parser. *International Journal on Digital Libraries*, 19(4):323–337, 2018.

[286] Animesh Prasad, Manpreet Kaur, and Min-Yen Kan. Neural parscit: a deep learning-based reference string parser. *International Journal on Digital Libraries*, pages 1–15, 2018.

[287] Juncheng Pu, Xiaodong Fu, Hai Dong, Pengcheng Zhang, and Li Liu. Dynamic adaptive federated learning on local long-tailed data. *IEEE Transactions on Services Computing*, 2024.

[288] Yanan Qian, Yunhua Hu, Jianling Cui, Qinghua Zheng, and Zaiqing Nie. Combining machine learning and human judgment in author disambiguation. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1241–1246, 2011.

[289] Yanan Qian, Qinghua Zheng, Tetsuya Sakai, Junting Ye, and Jun Liu. Dynamic author name disambiguation for growing digital libraries. *Information Retrieval Journal*, 18(5):379–412, 2015.

[290] Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*, 2024.

[291] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[292] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[293] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[294] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[295] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th*

Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789. Association for Computational Linguistics, 2018.

[296] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics, 2016.

[297] Shaina Raza, Shardul Ghuge, Chen Ding, Elham Dolatabadi, and Deval Pandya. Fair enough: Develop and assess a fair-compliant dataset for large language model training? *Data Intelligence*, 6(2):559–585, 05 2024.

[298] RDA FAIR Data Maturity Model Working Group. Fair data maturity model. 2020.

[299] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[300] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

[301] Roland Roller et al. A Medical Information Extraction Workbench to Process German Clinical Text.

[302] Matteo Romanello, Federico Boschetti, and Gregory Crane. Citations in the digital library of classics: extracting canonical references by using conditional random fields. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 80–87. Association for Computational Linguistics, 2009.

[303] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[304] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016.

[305] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT: A distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2020.

[306] Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. A review of author name disambiguation techniques for the pubmed bibliographic database. *Journal of Information Science*, 47(2):227–254, 2021.

[307] Anna Sauer, Shima Asaadi, and Fabian Küch. Knowledge Distillation Meets Few-Shot Learning: An Approach for Few-Shot Intent Classification Within and Across Domains. In Bing Liu, Alexandros Papangelis, Stefan Ultes, Abhinav Rastogi, Yun-Nung Chen, Georgios Spithourakis, Elnaz Nouri, and Weiyan Shi, editors, *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 108–119, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[308] Elyne Scheurwegs, Kim Luyckx, Léon Luyten, Walter Daelemans, and Tim Van den Bulcke. Data integration of structured and unstructured sources for assigning clinical codes to patient stays. *Journal of the American Medical Informatics Association*, 23(e1):e11–e19, 2016.

[309] Christophe Schinckus. The good, the bad and the ugly: An overview of the sustainability of blockchain technology. *Energy Research & Social Science*, 69:101614, 2020.

[310] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[311] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[312] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.

[313] Kristie Seymore, Andrew Mccallum, and Ronald Rosenfeld. Learning hidden markov model structure for information extraction. In *In AAAI 99 Workshop on Machine Learning for Information Extraction*, pages 37–42, 1999.

[314] Anmol Sharma, Sulayman K Sowe, Soo-Yon Kim, Sayed Hoseini, Fidan Limani, Zeyd Boukhers, Christoph Lange, and Stefan Decker. Fair data assessment using llms: The fair-way. In *Proceedings of the 34th ACM*

International Conference on Information and Knowledge Management, pages 5228–5232, 2025.

[315] Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. Towards automated icd coding using deep learning. *arXiv preprint arXiv:1711.04075*, 2017.

[316] Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A Smith, and Simon S Du. Decoding-time language model alignment with multiple objectives. *Advances in Neural Information Processing Systems*, 37:48875–48920, 2024.

[317] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[318] Stefano Silvestri, Francesco Gargiulo, Mario Ciampi, and Giuseppe De Pietro. Exploit multilingual language model at scale for icd-10 clinical text classification. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7, 2020.

[319] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[320] Neil R Smalheiser and Vetle I Torvik. Author name disambiguation. *Annual review of information science and technology*, 43(1):1, 2009.

[321] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[322] Kacper Sokol and Peter A Flach. Glass-box: Explaining ai decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In *IJCAI*, pages 5868–5870, 2018.

[323] Yanjie Song, Ponnuthurai Nagaratnam Suganthan, Witold Pedrycz, Junwei Ou, Yongming He, Yingwu Chen, and Yutong Wu. Ensemble reinforcement learning: A survey. *Applied Soft Computing*, 149:110975, 2023.

[324] Alan Souza, Viviane Moreira, and Carlos Heuser. Arctic: metadata extraction from scientific papers in pdf using two-layer crf. In *Proceedings of the 2014 ACM symposium on document engineering*, pages 121–130, 2014.

[325] Yash Srivastava, Vaishnav Murali, Shiv Ram Dubey, and Snehasis Mukherjee. Visual question answering using deep learning: A survey and performance analysis. *arXiv preprint arXiv:1909.01860*, 2019.

[326] Christopher Stahl, Steven Young, Dasha Herrmannova, Robert Patton, and Jack Wells. Deeppdf: A deep learning approach to extracting text from pdfs. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), 2018.

[327] Ewout W Steyerberg and Yvonne Vergouwe. Towards better clinical prediction models: seven steps for development and an abcd for validation. *European heart journal*, 35(29):1925–1931, 2014.

[328] Sebastian Stier, Arnim Bleier, Malte Bonart, Fabian Mörsheim, Mahdi Bohlouli, Margarita Nizhegorodov, Lisa Posch, Jürgen Maier, Tobias Rothmund, and Steffen Staab. Systematically monitoring social media: The case of the german federal election 2017. *arXiv preprint arXiv:1804.02888*, 2018.

[329] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[330] Chang Sun, Vincent Emonet, and Michel Dumontier. A comprehensive comparison of automated fairness evaluation tools. In *Semantic Web Applications and Tools for Health Care and Life Sciences*, volume 3127 of *CEUR Workshop Proceedings*, pages 44–53. Rheinisch-Westfaelische Technische Hochschule Aachen * Lehrstuhl Informatik V, January 2022. Publisher Copyright: Copyright © 2022 for this paper by its authors.; 13th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences, SWAT4HCLS 2022 ; Conference date: 10-01-2022 Through 14-01-2022.

[331] Xiaoling Sun, Jasleen Kaur, Lino Possamai, and Filippo Menczer. Detecting ambiguous author names in crowdsourced scholarly data. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 568–571. IEEE, 2011.

[332] Kenji Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017.

[333] Vighneswara Swamy and Munusamy Dharani. Investor attention using the google search volume index–impact on stock returns. *Review of Behavioral Finance*, 2019.

[334] Rajesh Talluri and Sanjay Shete. Using the weighted area under the net benefit curve for decision curve analysis. *BMC medical informatics and decision making*, 16(1):1–9, 2016.

[335] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1165–1174, 2015.

[336] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[337] Jie Tang, Alvis CM Fong, Bo Wang, and Jing Zhang. A unified probabilistic framework for name disambiguation in digital library. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):975–987, 2011.

[338] Kevin Ten Haaf, Jihyoun Jeon, Martin C Tammemägi, Summer S Han, Chung Yin Kong, Sylvia K Plevritis, Eric J Feuer, Harry J de Koning, Ewout W Steyerberg, and Rafael Meza. Risk prediction models for selection of lung cancer screening candidates: a retrospective validation study. *PLoS medicine*, 14(4):e1002277, 2017.

[339] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 580–599. Springer, 2020.

[340] Dominika Tkaczyk, Andrew Collins, Paraic Sheridan, and Joeran Beel. Machine learning vs. rules and out-of-the-box vs. retrained: An evaluation of open-source bibliographic reference and citation parsers. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pages 99–108. ACM, 2018.

[341] Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Lukasz Bolikowski. Cermine: Automatic extraction of structured metadata from scientific literature. *Int. J. Doc. Anal. Recognit.*, 18(4):317–335, 2015.

[342] Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(4):317–335, Dec 2015.

[343] Kristin M Tolle, D Stewart W Tansley, and Anthony JG Hey. The fourth paradigm: data-intensive scientific discovery [point of view]. *Proceedings of the IEEE*, 99(8):1334–1337, 2011.

[344] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[345] Hung Nghiep Tran, Tin Huynh, and Tien Do. Author name disambiguation by using deep neural network. In *Asian Conference on Intelligent Information and Database Systems*, pages 123–132. Springer, 2014.

[346] Michael Tsang, Hanpeng Liu, Sanjay Purushotham, Pavankumar Murali, and Yan Liu. Neural interaction transparency (nit): Disentangling learned interactions for improved interpretability. *Advances in Neural Information Processing Systems*, 31:5804–5813, 2018.

[347] Nicola Uras, Lodovica Marchesi, Michele Marchesi, and Roberto Tonelli. Forecasting bitcoin closing price series using linear regression and neural networks models. *PeerJ Computer Science*, 6:e279, 2020.

[348] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized collaborative learning of personalized models over networks. In *Artificial Intelligence and Statistics*, pages 509–517. PMLR, 2017.

[349] Ankit Vani, Yacine Jernite, and David Sontag. Grounded recurrent neural networks. *arXiv preprint arXiv:1705.08557*, 2017.

[350] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[351] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

[352] Andrew J Vickers and Angel M Cronin. Everything you always wanted to know about evaluating prediction models (but were too afraid to ask). *Urology*, 76(6):1298–1301, 2010.

[353] Guangyu Wang, Guoxing Yang, Zongxin Du, Longjun Fan, and Xiaohu Li. ClinicalGPT: Large Language Models Finetuned with Diverse Medical Data and Comprehensive Evaluation.

[354] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174*, 2018.

[355] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77:34–49, 2018.

[356] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models Are Zero-Shot Learners. *arXiv preprint arXiv:2109.01652*, 2022.

[357] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[358] Sascha Welten, Yongli Mou, Laurenz Neumann, Mehrshad Jaberansary, Yeliz Yediel Ucer, Toralf Kirsten, Stefan Decker, and Oya Beyan. A privacy-preserving distributed analytics platform for health care data. *Methods of information in medicine*, 61(S 01):e1–e11, 2022.

[359] Mark Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gaby Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Olavo Bonino da Silva Santos, Philip Bourne, Jildau Bouwman, Anthony Brookes, Tim Clark, Merce Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris Evelo, Richard Finkers, and Barend Mons. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3, 03 2016.

[360] Mark Wilkinson, Michel Dumontier, Susanna-Assunta Sansone, Luiz Olavo Bonino da Silva Santos, Mario Godoy, Dominique Batista, Peter McQuilton, Tobias Kuhn, Philippe Rocca-Serra, Merce Crosas,

and Erik Schultes. Evaluating fair maturity through a scalable, automated, community-governed framework. 05 2019.

[361] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

[362] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[363] Hao Wu, Bo Li, Yijian Pei, and Jun He. Unsupervised author disambiguation using dempster–shafer theory. *Scientometrics*, 101(3):1955–1972, 2014.

[364] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.

[365] Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, Yang Xiang, et al. Deep learning in clinical natural language processing: a methodical review. *Journal of the American Medical Informatics Association*, 2020.

[366] Yonghui Wu, Min Jiang, Jianbo Lei, and Hua Xu. Named entity recognition in chinese clinical text using deep neural network. *Studies in health technology and informatics*, 216:624, 2015.

[367] Yonghui Wu, Jun Xu, Yaoyun Zhang, and Hua Xu. Clinical abbreviation disambiguation using neural word embeddings. In *Proceedings of BioNLP 15*, 2015.

[368] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[369] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[370] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

[371] Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 649–658, 2019.

[372] Jun Xu, Siqi Shen, Dongsheng Li, and Yongquan Fu. A network-embedding based method for author disambiguation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1735–1738, 2018.

[373] Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Piyush Mathur, Frank Papay, Ashish K Khanna, Jacek B Cywinski, Kamal Maheshwari, et al. Multimodal machine learning for automated icd coding. In *Machine Learning for Healthcare Conference*, pages 197–215. PMLR, 2019.

[374] Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, 2018.

[375] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2025.

[376] Kai-Hsiang Yang and Yi-Hsuan Wu. Author name disambiguation in citations. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 335–338. IEEE, 2011.

[377] Liuqing Yang, Xiao-Yang Liu, Xinyi Li, and Yinchuan Li. Price prediction of cryptocurrency: An empirical study. In *International Conference on Smart Blockchain*, pages 130–139. Springer, 2019.

[378] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 21–29, 2016.

[379] Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. CrossFit: A Few-shot Learning Challenge for Cross-task Generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189. Association for Computational Linguistics, 2021.

[380] Ping Yin, Ming Zhang, ZhiHong Deng, and DongQing Yang. Metadata extraction from bibliographies using bigram hmm. In *International Conference on Asian Digital Libraries*, pages 310–319. Springer, 2004.

[381] Wang Yiying and Zang Yeze. Cryptocurrency price analysis with artificial intelligence. In *2019 5th International Conference on Information Management (ICIM)*, pages 97–101. IEEE, 2019.

[382] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2037–2045, 2021.

[383] Hubert Dariusz Zając et al. Ground truth or dare: Factors affecting the creation of medical datasets for training ai. In *Procs. of the 2023 AAAI/ACM AIES Conference*, page 351–362, 2023.

[384] Baichuan Zhang and Mohammad Al Hasan. Name disambiguation in anonymized graphs using network embedding. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1239–1248, 2017.

[385] Baichuan Zhang, Murat Dundar, and Mohammad Al Hasan. Bayesian non-exhaustive classification a case study: Online name disambiguation using temporal record streams. In *Proceedings of the 25th acm international on conference on information and knowledge management*, pages 1341–1350, 2016.

[386] Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan, and Ping Zhang. Combining structured and unstructured data for predictive models: a deep learning approach. *BMC medical informatics and decision making*, 20(1):1–11, 2020.

[387] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5014–5022, 2016.

[388] Quan-shi Zhang and Song-chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[389] Quanshi Zhang, Xin Wang, Ying Nian Wu, Huilin Zhou, and Song-Chun Zhu. Interpretable cnns for object classification. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3416–3431, 2020.

[390] X. Zhang, J. Zou, D. X. Le, and G. R. Thoma. A structural svm approach for reference parsing. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 479–484, Dec 2010.

[391] Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Interpretable visual question answering by visual grounding from attention supervision mining. In *2019 ieee winter conference on applications of computer vision (wacv)*, pages 349–357. IEEE, 2019.

[392] Yutao Zhang, Fanjin Zhang, Peiran Yao, and Jie Tang. Name disambiguation in aminer: Clustering, maintenance, and human in the loop. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1002–1011, 2018.

[393] Jianyu Zhao, Peng Wang, and Kai Huang. A semi-supervised approach for author disambiguation in kdd cup 2013. In *Proceedings of the 2013 KDD CUP 2013 Workshop*, pages 1–8, 2013.

[394] Zhengyun Zhao, Qiao Jin, Fangyuan Chen, Tuorui Peng, and Sheng Yu. A large-scale dataset of patient summaries for retrieval-based clinical decision support systems. *Scientific data*, 10(1):909, 2023.

[395] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019.

[396] Lingling Zhou, Cheng Cheng, Dong Ou, and Hao Huang. Construction of a semi-automatic icd-10 coding system. *BMC medical informatics and decision making*, 20(1):1–12, 2020.

[397] Dawei Zhu, Aditya Mogadala, and Dietrich Klakow. Image manipulation with natural language using two-sided attentive conditional generative adversarial network. *Neural Networks*, 136:207–217, 2021.

[398] Xi Zhu, Zhendong Mao, Chunxiao Liu, Peng Zhang, Bin Wang, and Yongdong Zhang. Overcoming language priors with self-supervised learning for visual question answering. *arXiv preprint arXiv:2012.11528*, 2020.

[399] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[400] Jie Zou, Daniel Le, and George R. Thoma. Locating and parsing bibliographic references in html medical articles. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(2):107–119, Jun 2010.