From Fine-tuning to Prompting: A Paradigm Shift in Knowledge Graph Construction

From Fine-tuning to Prompting: A Paradigm Shift in Knowledge Graph Construction

Xue Li

INVITATION

You are cordially invited
to the public defense of
my PhD thesis titled:

From Fine-tuning to Prompting: A Paradigm Shift in Knowledge Graph Construction

Xue Li

on Wednesday, 04 February
2026 at 10 am sharp.

The defense will be
followed
by a reception at the
same location.

The paranymfs:
Az Ali
Danru Xu

FROM FINE-TUNING TO PROMPTING: A PARADIGM SHIFT IN KNOWLEDGE GRAPH CONSTRUCTION

XUE LI

Cover and bookmark designed by Kasandra Poague.

SIKS Dissertation Series No. 2026-12
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The work was performed within the Intelligent Data Engineering Lab at the University of Amsterdam. Funding for this research comes from the Dutch Research Council (NWO) through grant MVI.19.032.

UNIVERSITY OF AMSTERDAM

From Fine-tuning to Prompting: A Paradigm Shift in Knowledge Graph Construction

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 4 februari 2026, te 10.00 uur

door Xue Li
geboren te Henan

Promotiecommissie

Promotor:	prof. dr. P.T. Groth	Universiteit van Amsterdam
Copromotor:	dr. rer. nat. J.C. Kalo	Universiteit van Amsterdam
Overige leden:	prof. dr. C. Monz	Universiteit van Amsterdam
	prof. dr. S. Milan	Universiteit van Amsterdam
	prof. dr. E. Simperl	King's College London
	prof. dr. K. Hose	TU Wien
	dr. L. Stork	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

ABSTRACT

Knowledge graphs (KGs) provide structured, machine-actionable representations of information that support search, reasoning, and decision-making. Constructing them, however, remains challenging in complex domains such as organizational conversations, where data is noisy, evolving, and context-dependent. This thesis examines how knowledge graph construction (KGC) can adapt to these conditions through two complementary perspectives: (i) analyzing the limitations of the pretrain-then-finetune (PTFT) paradigm when applied to conversational data, and (ii) exploring how the emerging pretrain, prompt, and predict (PPP) paradigm can provide more flexible and cost-efficient workflows. In the first part, we investigate the fragility of PTFT-based information extraction models under real-world variation. We show that distribution shifts in named entity recognition lead to large and predictable performance drops; that static topic models, though semantically coherent, struggle to detect the emergence of new topics; and that cross-document coreference in multi-party email exposes persistent weaknesses in current methods. These findings highlight the limits of task-specific models in domains shaped by input shifts, temporal change, and long conversational structure. In the second part, we turn to PPP-based workflows that leverage large language models through prompting rather than fine-tuning. We demonstrate that instruction-tuned LLMs can achieve competitive results in relation extraction, provided schema knowledge is carefully encoded. We introduce knowledge-centric prompt composition to guide in-context learning for knowledge base construction, showing that prompts enriched with schema constraints and examples substantially improve extraction quality. Finally, we propose a hybrid system for data preparation, TableSwift, which routes tasks between LLM-generated code and deterministic fallbacks to reduce costs while maintaining accuracy on transformation, error detection, and entity matching. Taken together, this thesis traces a critical paradigm shift in KGC: from PTFT pipelines reliant on specialized models, toward PPP workflows that are promptable, adaptable, and cost-aware. By diagnosing the weaknesses of PTFT and designing PPP-based solutions, it offers both empirical insights and practical architectures for building reliable knowledge graphs in complex, real-world domains.

ACKNOWLEDGMENTS

A PhD journey is a solitary process, one that demands a rigorous exploration of one's own capabilities and constraints. It is an exercise in humility as much as it is an academic endeavor. My PhD began in the middle of the pandemic in October 2020. I arrived in Amsterdam as a stranger to the city, and I still vividly remember my walks along the Ringvaart, often the only highlight of the day, where my breath would fog my glasses behind a mask, blurring the world around me. In those early days, research was a struggle conducted in the small radius of a student hotel room, where I grappled with research questions through the digital distance of a computer screen.

However, seasons eventually turned. The spring of 2022 brought more than just the return of the sun; it brought the arrival of my partner, Az, moving from a different continent to join me, and the lifting of restrictions. This shift from solitude to connection reminded me that while research is about quiet reflection, it flourishes in community. This journey would not have been possible without the support of my supervisors, colleagues, friends, and family.

First, I would like to express my deepest gratitude to my supervisors. Paul, thank you for your unwavering positivity; you are a great problem solver who frequently grounded my research and put my mind at ease during the most challenging phases of this research. Jan, though our collaboration was more brief, I truly valued the insight and energy you brought to our research together. I would like to thank the members of my thesis committee for their time, their careful reading of this manuscript, and their presence at my defense. Beyond my immediate supervision, I have been fortunate to work alongside a great group of senior staff. My thanks go to Sara, Jacobijn, Sebastian, Frank, Victoria, Hazar, Daphne, and Lise. Collectively, you have built an environment that is as intellectually stimulating as it is welcoming, and I am a better researcher for having been part of it.

I extend my sincere gratitude to my colleagues in INDELab, who reminded me daily that the people we work alongside are just as important as the research we conduct. James, thank you for the long walks and talks around the office, and for being such a truly supportive friend. Stefan, I am grateful for your persistent "peer pressure" that made sure we never missed a social event. Melika, I cherished our office conversations and the laughter we shared. My time in the lab was made inspiring by those who preceded me: Thiviyan, thank you for your openness and consistent efforts to connect people; Madelon, thank you for setting such an inspirational example of what a PhD journey can look

like; and Daniel, your passion and rigor for research have left a lasting impression on me.

The journey was also defined by food and sports. I am grateful to Pengyu for the Friday hotpots and our many cycling and bouldering sessions; and to Zeyu, for the incredible meals and for teaching me the "proper" way to enjoy Lanzhou Lamian. Erkan, thank you for your support on the bouldering wall, and Danru, for always lifting my spirits. Lucas, I remain impressed by your cycling commute from The Hague.

Even as the lab continues to grow, that same vibrant energy remains. Yichun, keep up with your dark humor, and thank you for pushing me and making me a much more responsible adult. David, your "double life" is truly inspiring; I am still searching for the secret to your endless energy. Antonis, I am grateful for the office banter, and Teresa, our hours-long conversations made even the longest afternoons fly by. Imane, it was always such a warm pleasure to catch up with you.

Finally, Fina, I value our collaboration and the unique perspective you brought from your background. Shubha, thank you for hosting us many times and making us feel at home. Brad, your boundless curiosity for technology and philosophy serves as an inspiring example of lifelong learning. To our past visitors, Valentina, Anca, Davide, Fan, and Haneen, the spirit of INDELab would not be the same without you.

I was also fortunate to complete an internship at MotherDuck, hosted within the CWI office. I would like to thank my teammates Till and Adithya for their collaboration. My gratitude also goes to my colleagues in Amsterdam, Jeewon, Niclas, Florian, Boaz, Jelte, and Philip, and to those further afield, Steph and Jordan, for making me feel so welcome. To the CWI DA group, Peter, Stefan, Lotte, Ila, Azim, Leonardo, Daniel, and Paul, thank you for welcoming me into a new community of which I am now a proud member. I am also glad to share this group with my newer colleagues, Daniel, Cornelius, Aécio, and Omid; I look forward to the work, and the table tennis matches, ahead within the group.

Beyond the office, the vibrant life of Amsterdam was made kinder and brighter by the friends who shared it with me. To Yan Meng, Ziwei Jin, Pramiti, Michele, Ciro, Sebastian, Bruno, Zeyou, Jiayu, Veis, Yoyo, and Michele, thank you for the laughter and for making this city feel like home.

To my childhood friends, 常长，连绍娟，陈月萌，刘星秀，和买迪, 感谢你们一直以来的支持，无论距离多远，你们的友谊始终是我力量的源泉。

To my parents, 范艳萍与李长在, 如果没有你们对我的信任和支持，我绝对无法完成今天的博士。妈妈，感谢你教会我如何更有同理心的看待这个世界。爸爸，感谢你教会我要持之以恒，一生只需要做好一件事的态度。

To my partner, Az, meeting you at the dawn of this journey was the greatest grace of all; 这一路，你是同行者，是知友，亦是我心定之处。 As we build our new "cave" together, I hope we finally overcome our biggest challenge -

tracking down that missing laundry bag, and that you find only peace in your own PhD journey.

Lastly, I thank the version of myself that persisted through the transit from Jiaozuo to Amsterdam. Though the fog was often thick, I am grateful for the tranquility it carries, leaving me with the space to transform solitary thoughts into the work you see here.

Effy (Xue Li)
Almere Poort, January 2026

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Knowledge graphs (KGs) are structured representations of information where real-world entities are modeled as nodes and their relationships as edges [82]. They provide a unified view of knowledge that is both human-interpretable and machine-actionable, enabling downstream applications such as question answering [85], search [187], recommendation [231], and decision support [76]. By linking diverse pieces of information into a coherent graph structure, KGs help uncover implicit connections, support reasoning, and enable better data integration across domains [33, 82, 169].

To obtain KGs, Knowledge Graph Construction (KGC) consists of a diverse set of tasks ranging from Information Extraction (IE) from unstructured data, followed by entity/schema alignment, and knowledge fusion with existing databases [218].

Among these, IE, the process of automatically identifying and structuring information from unstructured text, serves as the foundation for many KGC systems, upon which structured knowledge is built [90]. Without reliable extraction of entities, relations, and other semantic signals from raw data, the subsequent steps in KGC cannot operate effectively. In this thesis, we focus on the IE stage of the KGC pipeline [193], specifically on tasks that are essential for capturing both the structural and semantic dimensions of unstructured data, as well as the entity alignment and data cleaning step when integrating to existing knowledge bases.

To build a comprehensive view of knowledge extraction, this thesis focuses on three IE tasks, four data cleaning tasks, and one thematic extraction task that together construct structured representations from unstructured data. Named Entity Recognition (NER), discussed in Chapter 2, identifies key entities that serve as the nodes of a knowledge graph, while Relation Extraction (RE) in Chapter 6 defines the edges that connect them. Coreference Resolution (Coref), discussed in Chapter 4, identifies if two entities are referring to the same real-world entity across documents to prevent graph fragmentation. Data preparation tasks, such as Entity Matching (EM), Error Detection (ED), Data Transformation (DT), and Data Imputation (DI) presented in Chapter 7, further clean and enrich the graph by detecting errors, imputing missing values, transforming and matching entities from heterogeneous sources. Finally, Topic Extraction, explored in Chapter 3, reveals the thematic structures that organize textual collections, complementing the entity- and relation-centric view of IE [95].

These tasks are essential for building coherent KGs, particularly when integrating information from diverse, noisy, or partially structured sources [179].

Finally, topic extraction provides higher-level semantic information, enabling KGs to support not only factual queries but also organization-level reasoning.

IE has long relied on carefully designed pipelines of task-specific models: one for recognizing named entities, another for linking them, yet another for disambiguating references. Over time, these pipelines evolved from rule-based systems [3, 64, 181] and feature-engineered classifiers [51, 167] to neural models trained end-to-end on large annotated corpora [98, 225]. The introduction of distributed representations [80], such as word embeddings [14, 144, 160], further enabled shared representations across tasks. This shift led to the rise of pretrained language models [207], which encode broad linguistic and factual knowledge from large text corpora [10, 131]. These neural models are typically trained under the *pretrain-then-finetune* (PTFT) paradigm, where large language models like BERT [54] are fine-tuned for downstream tasks. This paradigm, while effective in curated domains, breaks down in complex real-world settings where data is noisy, evolving, and structurally diverse [75, 192].

Recent advances in large language models (LLMs) have introduced a new paradigm: *pretrain, prompt, and predict* (PPP) [132]. Rather than fine-tuning a model per task, LLMs are now prompted to perform diverse tasks zero-shot or few-shot using natural language instructions. This shift radically simplifies the architecture of information extraction and opens up new opportunities for flexible, domain-agnostic workflows.

This thesis explores these two paradigms for KGC. We first study the limitations of PTFT pipelines in domains where data is sparse, noisy, unstructured, or ever-changing, what we refer to as **complex domains**. Then, we explore how PPP-based approaches can overcome these limitations. The thesis is structured in two parts:

- **Part I: Identifying Challenges of the Pretrain-then-Finetune Paradigm for Knowledge Graph Construction.** We analyze where and why PTFT-based Information Extraction (IE) models fail, focusing on three core challenges: distribution shifts in Named Entity Recognition (NER), topic emergence in dynamic corpora, and coreference in long-form conversations.
- **Part II: Designing Pretrain, Prompt, Predict-based Workflows for Knowledge Graph Construction.** We explore how LLMs can be prompted to perform KG construction tasks, including relation extraction, triple generation, and various data cleaning tasks, without the need for task-specific training, and propose architectures that maximize both flexibility and cost efficiency.

Figure 1.1 illustrates this paradigm shift. In the PTFT workflow, data must flow through several task-specific models, often trained and evaluated separately. By contrast, PPP allows us to encode tasks and structure directly into prompts, simplifying the process while enabling generalization across tasks and domains.

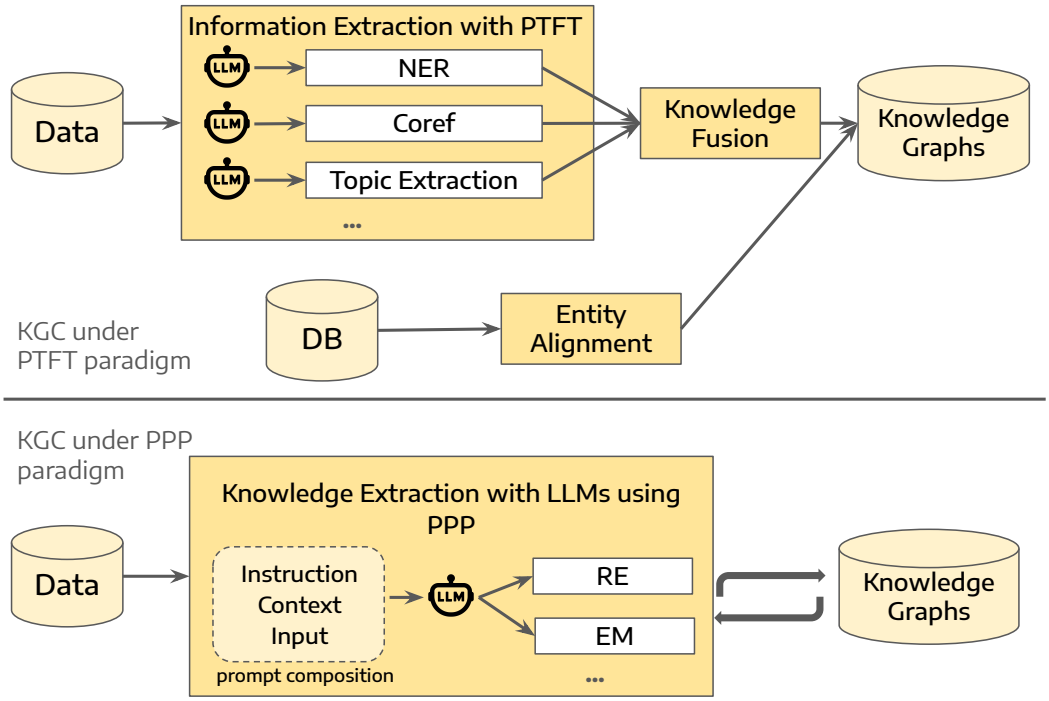

Figure 1.1: Two paradigms for constructing knowledge graphs.

1.1 IDENTIFYING CHALLENGES OF THE PRETRAIN-THEN-FINETUNE PARADIGM FOR KNOWLEDGE GRAPH CONSTRUCTION

In Part I of this thesis, we investigate the limitations of the PTFT paradigm in complex domains, specifically **conversational data** such as internal emails and organizational discussions. Rather than treating a domain as a fixed corpus or dataset, we adopt the variety space perspective, where a corpus is viewed as a subregion of a high-dimensional latent space [162]. This variety space captures the underlying linguistic and annotation-related factors, such as genre, sub-domain, socio-demographic variation, style, or annotator bias that shape the data distribution [170]. From this perspective, conversational corpora reflect a particularly challenging subspace, marked by informal syntax, context dependence, and discourse structure, all of which can potentially negatively impact the generalization capacity of the pretrained model.

This thesis focuses on conversational data because it captures the process behind the decision-making [205], not just its outcomes. Unlike finalized texts, such as published articles or technical standards, conversational data reveals the rich, often messy path towards consensus. It contains valuable information and insights into how decisions are made, such as what alternatives were discussed and how key actors influenced the direction. For example, public email

threads during the development of a standard can expose the reasoning and collaboration behind each decision [237], which are insights lost in the finalized documents. Studying such data is crucial for understanding the dynamics of institutional decision-making.

To understand how the PTFT paradigm performs in such settings, we systematically evaluate its effectiveness on key IE tasks, which are often handled through a series of task-specific models individually, before storing extracted knowledge into meaningful graphs.

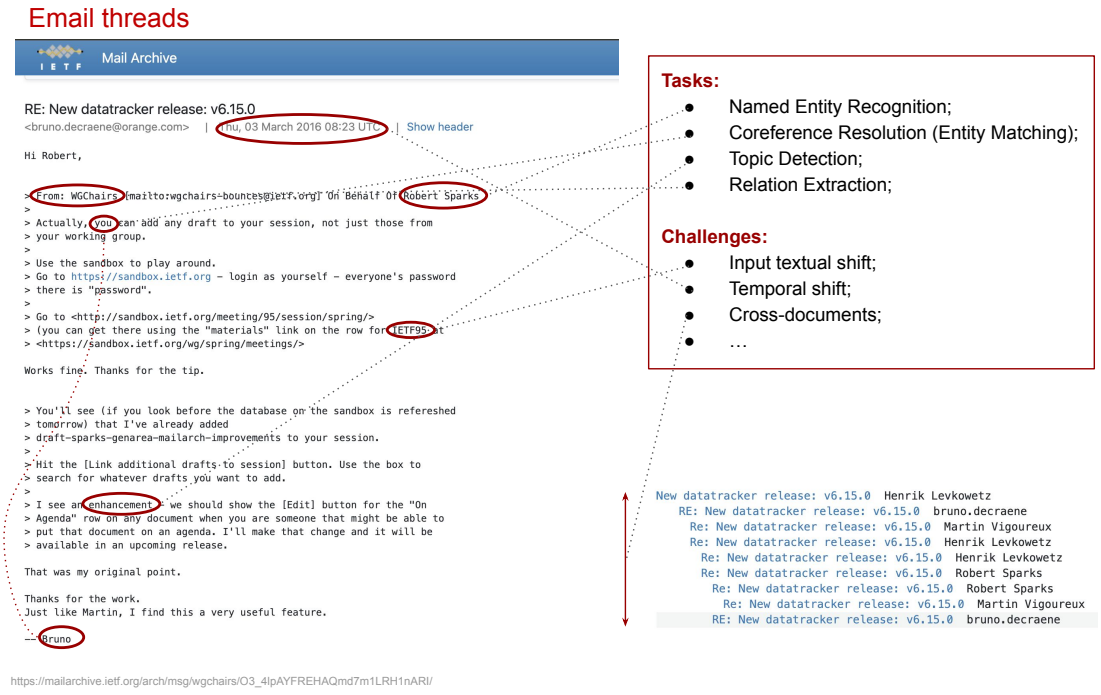

Figure 1.2: One example characterizing complex conversations.

To illustrate the challenges of IE under the PTFT paradigm, Figure 1.2 presents a real-world email thread from an Internet Engineering Task Force (IETF) working group mailing list[1]. This example email thread captures a multi-party discourse characterized by informal syntax, fragmented structure, and implicit references that pose challenges for existing IE approaches.

This short thread contains many of the complexities that arise in KGC workflows. To extract structured knowledge from this example, a typical PTFT pipeline may execute the following steps involving individual models for each step:

- **Named Entity Recognition (NER):** The extractor should be able to identify entities such as **Robert Sparks** and **Bruno** as persons.

1 The email example is from:`https://mailarchive.ietf.org/arch/msg/wgchairs/O3_4lpAYFREHAQmd7m1LRH1nARI/`

- **Coreference Resolution (CR):** The pipeline must resolve references across the conversation, such as linking the mention **you** to its antecedent **Bruno** as in Figure 1.2.
- **Topic Detection (TD):** Long email threads often involve multiple intertwined topics. Identifying thematic topics from multi-topic texts enables the construction of topic-specific knowledge graphs.
- **Relation Extraction (RE):** Once entities are identified, the system must extract the semantic relations between them, which form the edges of the knowledge graph.

These tasks become significantly harder in complex domains due to the following key challenges:

- **Input Shift:** Domain-specific terms such as **IETF95** or abbreviations not seen in training data cause NER models to fail.
- **Temporal Shift:** Conversations often evolve. In this example, the topic shifts mid-thread from current release issues to future enhancements. Topic detection models need to account for this temporal drift.
- **Cross-Documents Structure:** Conversations span multiple turns and speakers, often across long emails or threads. This challenges models with limited input windows and makes tasks like cross-document coreference particularly difficult. For example, resolving that **Martin** and **Martin Vigoureux**, as shown in Figure 1.2, refers to the same entity, which may require attending to emails several turns apart.

To this end, we identify tasks and challenges needed when use the PTFT paradigm for KGC.

These findings highlight the limitation of PTFT-based pipelines when deployed in real-world, complex settings. This motivates the second part of the thesis, where we turn to designing flexible solutions based on the PPP paradigm.

1.2 DESIGNING PRETRAIN, PROMPT AND PREDICT-BASED SOLUTIONS FOR KNOWLEDGE GRAPH CONSTRUCTION

The challenges identified in Part I largely stem from the limited generalization capabilities of task-specific models. While these models still represent the state of the art for many isolated tasks, their need for a large amount of task-specific training data to adapt to new tasks and domains significantly constrain their applicability in real-world KGC workflows.

The emergence of the PPP paradigm, driven by significantly larger pretrained language models [18, 36, 154], has expanded the range of tasks that can be addressed with minimal supervision due to their scaling laws [97]. These models often generalize better across tasks and domains, and can implicitly mitigate many of the challenges described in Part I, due to their ability to adapt to new domains in low-resource settings [18], whether the shift involves new tasks,

topics, or unfamiliar styles [18]. Additionally, the increasing context length of LLMs enables them to handle longer-form inputs [57], without requiring task-specific training or fine-tuning.

This shift from PTFT to PPP re-frames the KG construction process [157, 158]. Rather than relying on pipelines of specialized models, we instead provide LLMs with natural language prompts that encode task definitions, schema constraints, and example inputs. These prompts serve as a new interface for KG construction, configurable, interpretable, and portable across domains. For instance, instead of training a relation extraction model for each schema, we provide the model with a textual instruction such as: *"Extract all relations of type [person, worksAt, organization] from the text below."*

A particularly promising feature of this approach is *in-context learning* [58], where few-shot examples are embedded directly into the prompt to guide the model's behavior. These examples can be manually selected, randomly sampled, or retrieved dynamically using retrieval-augmented generation (RAG) [61, 114]. In the context of KGC, RAG allows the model to ground its predictions in task-relevant demonstrations or factual knowledge, significantly improving consistency [230].

However, this new paradigm introduces a different set of challenges. Specifically, the question is no longer *can* LLMs perform the task, but rather *how* to orchestrate them effectively: How can we design prompts that yield accurate outputs? How do we balance cost and performance? How do we use smaller models for specific tasks?

In Part II of this thesis, we explore these orchestration challenges across multiple KG-related tasks, including relation extraction, and data preparation, and propose prompting-based workflows that are both flexible and cost-efficient.

1.3 RESEARCH QUESTIONS

This thesis is structured around five research questions, each addressed in a dedicated chapter. The first three questions correspond to Part I, which focuses on identifying the limitations of the PTFT paradigm in complex domains. The last two questions correspond to Part II, where we explore prompting-based solutions for KGC.

In Part I, we examine how the PTFT paradigm struggles along three key dimensions in the context of KGC for complex conversations: input distribution shift, temporal evolution, and cross-document structure.

We formulate our first research question as follows:

RQ1 *How do pretrained-then-finetuned Named Entity Recognition models perform under distribution shifts?*

This research question addresses the situation where the source and target domains differ. Such distribution shifts can significantly degrade the perfor-

mance of models fine-tuned on one domain and applied to another. However, existing work does not offer a systematic approach for measuring these shifts or understanding their explicit impact on downstream model performance.

To address this question, we take the widely adopted pre-trained BERT [54] model and fine-tune it for the NER task under various types of distribution shifts. Specifically, we test two types of shifts, including input shift and label shift, across three representations, including frequency-based and embedding-based representations, on 12 benchmark NER datasets. We quantify performance degradation and analyze how it correlates with measurable aspects of domain shift. Our results show that models perform well when fine-tuned and tested on the same dataset, but experience dramatic performance drops when evaluated on out-of-domain data.

These findings suggest that measuring distribution shifts between the source and target domains can help anticipate model performance degradation and inform the amount of data needed for effective fine-tuning. In the context of complex conversations, we observe a similarly sharp drop in performance when models are transferred across datasets.

Having explored domain-level variation, we next investigate a second dimension of complexity: **temporal evolution**. Specifically, we ask:

RQ2 *To what extent can pre-trained topic models perform topic emergence detection?*

One defining characteristic of KGC for complex domains is its ever-evolving nature. Within organizational discourse, topics shift frequently, and new themes can emerge at any point. We treat topic extraction as a form of document-level information extraction, aiming to uncover latent thematic structures rather than explicit entities or relations.

While BERT-based models perform well on many semantic tasks, including topic extraction, their ability to detect the emergence of new topics over time remains unclear. This is in part due to the lack of evaluation metrics specifically designed to measure temporal sensitivity in topic models. To address this gap, we compare the performance of the neural topic model BERTopic with classical approaches such as CoWords Clustering and Latent Dirichlet Allocation (LDA). We introduce a new metric for evaluating topic emergence detection in an unsupervised setting, allowing for a fair comparison across modeling paradigms. Our results show that while BERTopic performs well at semantic grouping, it falls short in detecting emerging topics compared to classical non-neural methods.

These findings suggest that topic models require improved temporal modeling capabilities to be effective in dynamic domains like conversations.

Having now examined the challenges introduced by domain shift and temporal drift, we turn to a third dimension of complexity: long-form, cross-document structure. As shown in Figure 1.2, complex conversations often unfold over extended email threads with multiple participants and references that span sev-

eral messages. To understand how this structure affects model performance, we ask:

RQ3 *What are the challenges for pretrained-then-finetuned models for cross-document coreference resolution in complex conversations?*

While BERT-based models have shown improved performance in coreference resolution, the cross-document setting remains particularly challenging. In this chapter, we finetuned an end-to-end framework that utilizes spanBert for cross-document coreference resolution using public emails. Our analysis reveals that frequent shifts in pronoun usage, informal language, varied surface forms, and entity sparsity contribute to a significant drop in performance when applied to complex conversations. We categorize these difficulties and propose six directions for improving cross-document coreference in future work.

Together, these findings highlight the fundamental limitations of task-specific, PTFT models in KGC for complex domains. Across the dimensions of distribution shift, temporal evolution, and long-form document structure, such models struggle to generalize effectively without domain-specific adaptation. As a result, constructing knowledge graphs in these settings often requires extensive manual intervention, task-specific engineering, or retraining. These limitations make it challenging to build high-quality knowledge graphs efficiently in complex domains.

These insights motivate the second part of this thesis, where we turn our attention to designing prompting-based workflows for the knowledge graph construction, focusing on how to orchestrate LLMs effectively to balance accuracy, generalizability, and cost.

* * *

Most of the challenges identified in Part I stem from models overfitting to their training data, resulting in poor generalization to new domains, time periods, or document structures. The emerging capabilities of PPP-based workflows offer a compelling alternative by enabling flexible, task-agnostic performance through prompting.

In Part II of this thesis, we first examine how a PPP-based approach can support knowledge base construction, particularly using in-context learning. We enhance prompting by selecting relevant demonstration examples and augmenting them with auxiliary information from Wikidata. Our method outperforms strong baselines by over 40 points, demonstrating that PPP-based pipelines are a promising direction for knowledge graph construction.

While PPP-based workflows show strong performance, relying on large proprietary models [154] has several drawbacks. These include limited control over the model's behavior [154, 189], inability to tailor it to specific domain needs due to the model's size and deployment constraints, and the requirement to send data to third-party servers [220].

This raises the question: can smaller, open-source models be effectively leveraged in a PPP workflow for tasks like relation extraction? To investigate this, we ask:

RQ4 *To what extent can we improve the ability of Pretrain, Prompt and Predict-based models to perform Relation Extraction?*

Due to computational constraints, it is often impractical to fine-tune LLMs by updating all parameters. However, recent work on parameter-efficient fine-tuning has made it feasible to adapt LLMs in a more budget-conscious manner. Our results show that instruction-tuned LLMs can achieve performance comparable to smaller, fully supervised models. Interestingly, we also observe that LLMs can generate plausible triples that are not present in the gold labels, suggesting a need for evaluation methods beyond exact-match metrics.

While both in-context learning and instruction-tuning demonstrate impressive capabilities for knowledge graph construction, they come with trade-offs. In-context learning with large models is often costly and less controllable, while smaller instruction-tuned models may struggle with generalization. A promising middle ground is to develop systems that strategically orchestrate the use of LLMs, balancing generalization with cost-efficiency.

To explore this, we pose our final research question:

RQ5 *How can PPP workflow improve the cost-efficiency for data preparation tasks?*

Prompting LLMs to process every data point for every task can be prohibitively expensive, particularly for repetitive operations. One promising direction is to use LLMs strategically as *decision-makers* rather than direct executors of all subtasks. To this end, we design a system that dynamically decides whether to generate transformation code via LLMs or fall back on rule-based alternatives, particularly reserving LLM calls for semantically challenging cases. This method design principle facilitates efficient KGC pipelines, while large volumes of data can be processed with deterministic rules, and complex cases such as entity matching or domain-specific relation extraction can benefit from LLM's semantic reasoning capabilities.

Our results show that this hybrid framework achieves state-of-the-art performance on data transformation tasks, highlighting its suitability for practical use. Moreover, in tasks such as error detection, entity matching, and data imputation, the system substantially reduces cost while maintaining high accuracy, outperforming baseline methods on multiple metrics.

Through this final research question, we demonstrate that PPP workflows can be orchestrated in a cost-aware and task-adaptive manner.

To conclude, this thesis identifies key challenges for knowledge graph construction in complex domains and proposes dynamic, LLM-based solutions that leverage prompting effectively. In the next section, we summarize the main contributions of the work.

1.4 MAIN CONTRIBUTIONS

This thesis advances the field of KGC by identifying the limitations of task-specific pipelines in complex domains and proposing prompt-driven workflows leveraging LLMs. The contributions span four categories: empirical, algorithmic, theoretical, and resource-based.

Empirical Contributions

- **Systematic evaluation of distribution shifts in NER** [Chapter 2]: We quantify how PTFT models degrade under varying levels of domain shift across 12 benchmark NER datasets, and show how shift metrics can predict performance loss.
- **Evaluation of topic models for topic emergence detection** [Chapter 3]: We introduce an unsupervised metric to assess a model's ability to detect emerging topics and show that BERT-based models underperform compared to classical baselines on this task.
- **Empirical study of cross-documents coreference resolution in long-form conversations** [Chapter 4]: We identify core challenges in cross-document coreference resolution, including pronoun ambiguity, sparse mentions, and informal discourse, and provide an analysis of failure modes specific to conversational data.
- **Evaluation of instruction-tuned LLMs for Relation Extraction** [Chapter 6]: We evaluate instruction-tuned LLMs on relation extraction and show that standard exact-match metrics fail to capture the quality of open-ended LLM outputs, compared to human evaluation. Besides, our results also show that these models encode rich parametric knowledge capable of generating out-of-scope triples.

Algorithmic Contributions

- **Prompt-based pipeline for knowledge base construction** [Chapter 5]: We design a PPP-based workflow using in-context learning with retrieved examples and external Wikidata knowledge, achieving large gains over baselines in triple extraction.
- **Hybrid system for cost-efficient LLM orchestration** [Chapter 7]: We introduce a system that dynamically chooses between LLM-generated code and lightweight fallbacks for data wrangling tasks, demonstrating strong performance on transformation, error detection, and entity matching with reduced cost.

Resource Contributions

- **Benchmark framework and reproducible evaluations** [Chapters 2, 3, 7]: We release datasets, shift metrics, topic emergence evaluation code, and prompt-based pipelines to support reproducibility and future research in KGC.

1.5 THESIS OVERVIEW

The structure of the chapters is shown in Figure 1.3. The thesis is composed of two parts, and each part consists of three chapters. Chapters 2, 3, and 4 are included in Part I, identifying the challenges when using task-specific models. Chapters 5, 6, 7 belong to Part II, designing flexible LLM-based workflows. Each chapter is addressing a research question described in Section 1.3.

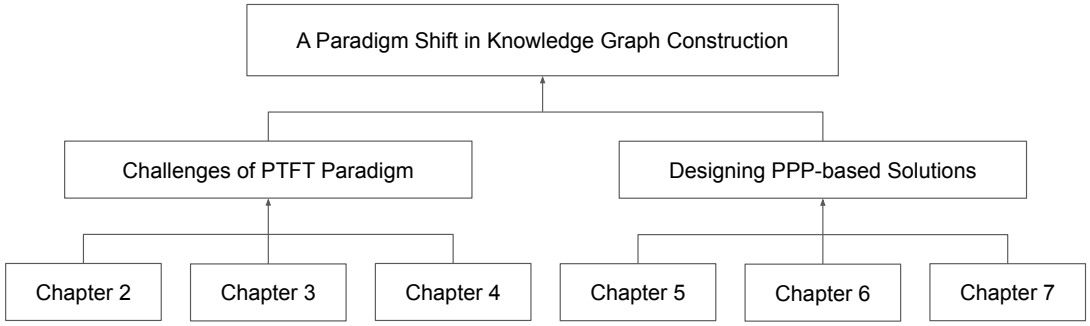

Figure 1.3: The overall structure of the thesis, with their corresponding parts.

1.6 ORIGINS

This thesis is based on six research papers, each of which forms the basis of a chapter. The corresponding venues and author contributions are listed below. Author roles are described using the CRediT taxonomy [148].

- **Chapter 2** is based on the paper: **Xue Li** and Paul Groth. *"How different is different? Systematically identifying distribution shifts and their impacts in NER datasets"* [121]. In: *Language Resources and Evaluation Journal*, Springer, 2024.
  XL: Conceptualization, Methodology, Software, Writing - Original Draft. PG: Conceptualization, Supervision, Writing - Reviewing and Editing.
- **Chapter 3** is based on the paper: **Xue Li**, Ciro D Esposito, Paul Groth, Jonathan Sitruk, Balazs Szatmari, and Nachoem Wijnberg. *"Evaluation of Unsupervised Static Topic Models' Emergence Detection Ability"* [120]. *PeerJ Computer Science*, PeerJ, 2025.
  XL: Conceptualization, Methodology, Software, Data Curation, Formal Analysis, Visualization, Writing – Original Draft, Writing – Review & Editing.

CE: Conceptualization, Methodology, Formal Analysis, Writing – Review & Editing. PG, JS, BS, NW: Conceptualization, Writing – Review & Editing.

- **Chapter 4** is based on the paper: **Xue Li**, Sara Magliacane, Paul Groth. *"The challenges of cross-document coreference resolution for email"* [123]. In: *K-CAP 2021: the 11th International Conference on Knowledge Capture*.
  XL: Conceptualization, Data Curation, Analysis, Writing – Original Draft. SG: Supervision, Writing – Review & Editing. PG: Supervision, Writing – Review & Editing.
- **Chapter 5** is based on the paper: **Xue Li**, Anthony Hughes, Majlinda Llugiqi, Fina Polat, Paul Groth, Fajar J Ekaputra. *"Knowledge-centric Prompt Composition for Knowledge Base Construction from Pre-trained Language Models"* [122]. In: *ISWC 2023 LM-KBC: the 1st challenge of Language Models for Knowledge Base Construction at the 22nd International Semantic Web Conference*.
  XL: Conceptualization, Methodology, Investigation, Software, Writing – Original Draft. AH, ML, FP: Conceptualization, Methodology, Investigation, Software, Writing – Review & Editing. FE: Conceptualization, Methodology, Investigation, Software, Supervision, Writing – Review & Editing. PG: Supervision, Writing – Review & Editing.
- **Chapter 6** is based on the paper: **Xue Li**, Fina Polat, Paul Groth. *"Do Instruction-tuned Large Language Models Help with Relation Extraction?"* [124]. In: *ISWC 2023 LM-KBC: the 1st Workshop of Knowledge Base Construction from Pre-Trained Language Models at the 22nd International Semantic Web Conference*.
  XL: Conceptualization, Methodology, Software, Writing – Original Draft. FP: Conceptualization, Methodology, Software, Writing – Review & Editing. PG: Conceptualization, Writing – Review & Editing
- **Chapter 7** is based on the following two papers: **Xue Li**, Till Döhmen. *"Towards Efficient Data Wrangling with LLMs using Code Generation"* [119]. In: *SIGMOD'24 DEEM: Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning at the 2024 ACM International Conference on Management of Data*. XL: Conceptualization, Methodology, Software, Writing – Original Draft. TD: Conceptualization, Methodology, Writing – Review & Editing.
  Xue Li, Till Döhmen, Jan-Christoph Kalo, Paul Groth. *"TableSwift: Efficient Data Wrangling with Large Language Models using Code Generation"*. Under submission.
  XL: Conceptualization, Methodology, Software, Writing – Original Draft. TD: Conceptualization, Methodology, Writing – Review & Editing. JK: Conceptualization, Writing – Review & Editing. PG: Supervision, Writing – Review & Editing.

1.7 RESEARCH DATA AND SOFTWARE

All code and data used in this thesis are publicly available to support reproducibility and future research. Links to code repositories and data sources are organized by chapter.

Code Repositories

- **NER Distribution Shift (Chapter 2):** `https://github.com/effyli/dish-vm`
- **Topic Emergence Detection (Chapter 3):** `https://zenodo.org/records/14503316`
- **Cross-Document Coreference Resolution (Chapter 4):** `https://github.com/effyli/cdcr`
- **In-Context Learning for KBC (Chapter 5):** `https://github.com/effyli/lm-kbc`
- **Instruction-Tuned LLMs for RE (Chapter 6):** `https://github.com/INDElab/KGC-LLM`
- **TableSwift for Data Wrangling (Chapter 7):** `https://github.com/effyli/TableSwift`

Part I: Identifying Challenges of the Pretrain-then-Finetune Paradigm for Knowledge Graph Construction.

SUMMARY OF PART I

In Part I, we examine the limitations of the Pretrain-then-Finetune (PTFT) paradigm for Knowledge Graph Construction in complex domains through three tasks: Named Entity Recognition (NER), topic emergence detection, and cross-document coreference resolution. We demonstrate that task-specific models struggle with generalization in the presence of distribution shifts, temporal shift, and long-form, informal texts. These findings highlight the brittleness of static, monolithic pipelines in evolving real-world contexts and motivate the need for more flexible, adaptable approaches to information extraction. In the next part of the thesis, we explore how the emerging Pretrain, Prompt, and Predict (PPP) paradigm, enabled by large language models, can address these challenges and re-design the way we approach KGC.

2

SYSTEMATICALLY IDENTIFYING DISTRIBUTION SHIFTS AND THEIR IMPACTS IN NER DATASETS

To understand the limitations of the Pretrain-then-Finetune (PTFT) paradigm, we begin by investigating how distribution shifts affect model performance in the context of Named Entity Recognition (NER), a foundational task in knowledge graph construction. Distribution shifts occur when the test distribution diverges significantly from the training data, often leading to degraded performance and exposing a model's lack of generalization. Therefore, we ask the following research question:

RQ1 *How do pretrained-then-finetuned NER models perform under distribution shifts?*

In this chapter, we characterize two types of shifts: input shift and label shift. We then systematically evaluate BERT-based NER models across 12 benchmark datasets to assess how these shifts impact performance and where the PTFT paradigm begins to break down.

2.1 INTRODUCTION

Differences between training and inference distributions are a common occurrence in the field of Natural Language Processing (NLP). This difference can be observed, for instance, when the input data undergoes changes over time or when a model is employed on data from a new domain. This is known as *distribution shift* [164]. Consider the following example from Named Entity Recognition (NER):

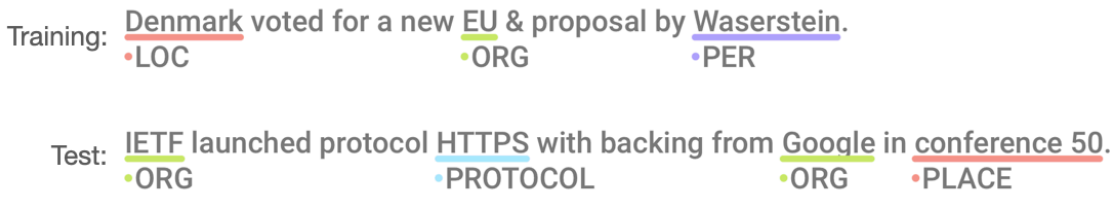

Figure 2.1: An example of differences in sentences that would be affected by a shift in the distribution of the NER training data.

The example shows two phenomena. First, the entities in the training example tend to be relatively well-known entities (e.g. EU), which are highly probable to be present in the data sources utilized by pre-trained language models [54, 109] that are widely used for NLP tasks. Conversely, the entities in the

inference example are unique to a particular domain (e.g. IETF). Second, the labels in the training example differ from the ones in the inference example. This is because entities from different domains possess different types, such as “Organization" versus “Protocol", and variations in labelling for the same type, such as “Location" and "Place". These phenomena embody two common shifts in NLP: input distribution shifts and label distribution shifts [164]. We show how these two types of shifts can affect the performance of an existing classifier with a toy example in Figure 2.2. The example shows when the test distribution differs from the training distribution, often caused by the change of the underlying relationship between the input x and the label y. When shifts happen, often the performance of the pre-trained classifier (shown in dashed lines) will no longer hold. In this chapter we primarily focus on category shift in the label space [112]. Despite the substantial body of literature on measuring domain similarity [46], detecting *when* a shift occurs remains a challenging task in the field. This task is known as *shift detection*.

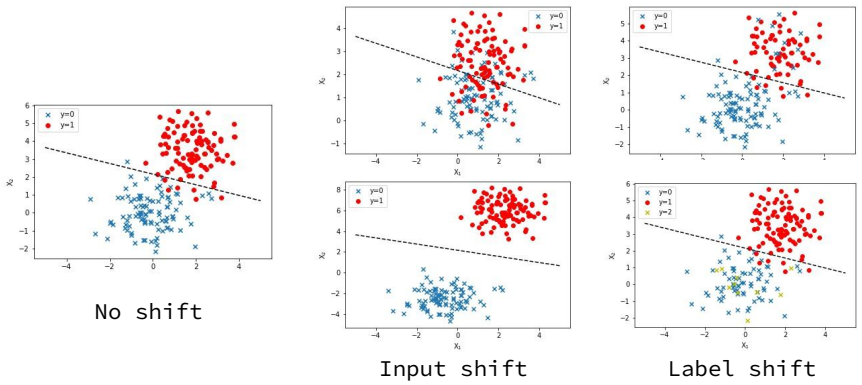

Figure 2.2: Toy examples of input shift and label shift. The dashed lines indicate an existing machine-learning classifier that performs well at training time. We show two possible scenarios when the relationship of x and y changes for each type of shift.

A key area where shift detection is useful is *domain adaptation*, which aims at adapting a model in the presence of distribution shifts [44]. One of the common supervised approaches to achieve adaptation is fine-tuning deep neural networks [203]. While fine-tuning can be effective, there is still a cost, such as determining the required amount of additional data for fine-tuning. To inform this decision, shift detection methods are frequently employed in other areas that employ machine learning [41, 106, 166]. This chapter frequently adopts statistical hypothesis testing as an underlying principled approach to the problem [41, 166]. Statistical two-sample testing is a methodology for determining whether the distribution of the training data p is equivalent to the distribution of the test data q. While this approach has been explored for computer vision tasks involving high-dimensional data, it has seen limited application to NLP

Hence, to better inform these decisions and quantify the potential impact of distribution shifts, this chapter undertakes a systematic investigation of shifts across benchmark corpora using statistical tests, which have been widely adopted for shift detection in the context of other machine learning tasks. In this chapter, we specifically focus on the NER task and detect distribution shifts across 12 different datasets that are representative of various domains. We use word frequency and sentence-level representations to characterize input distributions, and label frequency to characterize label distributions. Appropriate statistical tests are identified for each representation and employed to detect and quantify shifts. We then investigate the impact of domain shift in both the input and label space on performance in the supervised setting. We establish a relationship between the shift distance and the performance degradation. These results provide insights into what statistical test one needs to perform to make such a determination.

Summarizing, the contributions of this chapter are as follows:

- The systematic measurement of distribution shift between 12 NER benchmark datasets covering multiple domains.
- Systematic measurement of how much a distribution shift impacts performance for NER, a prototypical NLP task.
- Evidence that sentence-based representations provide better information for shift detection for the NER task.

2.2 RELATED WORK

Distribution shifts are prominent in real-world applications [60, 142, 173, 190], leading to growing interest in detecting them for machine learning tasks [41, 106, 166].

SHIFT TYPES In the broader landscape of machine learning, Wiles et al. [221] conducted a fine-grained analysis of distribution shifts, classifying them as spurious correlation, low data drift, and unseen data shift. Additionally, they evaluated 19 different methods on both synthetic and real-world datasets for vision tasks.

THE USE OF STATISTICAL TESTS The use of statistical tests for dataset shift detection was brought to the fore by Rabanser, Günnemann, and Lipton [166]. In their work, they developed a dataset shift detection framework which contains a dimensionality reduction component and a two-sample-testing component. They investigated multiple combinations of methods for each component, and tested on artificially generated covariates and label distribution shifts. Recently, based on two-sample tests for shift detection, Cobb and Van Looveren [41] developed a general drift detection framework borrowing machinery from

Corpus	Year	Document source	Domain	# Types	Category
GUM	2017	Wiki-family	Various	11	Wiki data
Wikigold	2009	Wikipedia text	Various	4	Wiki data
BTC	2016	Twitter	Mainstream news	3	Informal text
W-NUT17	2017	User-generated text	Various	6	Informal text
CEREC	2005	Informal emails	Work	4	informal text
AnEM	2012	Anatomical text	Anatomy	11	Specific field
i2b2-06	2006	Clinical text	Biomedical	7	Specific field
SEC-Filings	2015	Electronic filings	Finance	4	Specific field
SciERC	2018	Scientific abstracts	Scientific	6	Specific field
Re3d	2018	Documents related to defense and security analysis	Conflict in Syria and Iraq	10	Specific field
CoNLL-03	2003	Reuters news	Mainstream news	4	News
OntoNotes	2007-2012	Magazine, news, web, tele, etc	Various	18	General

Table 2.1: List of annotated datasets for English NER from different domains.

causal inference. The framework is used to deal with the situation when the inference data are not expected to form an i.i.d. sample from the historical data distribution.

DOMAIN SIMILARITY Within the field of NLP, researchers have explored various methods for measuring domain similarity in the context of domain adaptation including using target vocabulary covered rate and language model perplexity[46]. However, these methods work well under the assumption that there are sufficient data from the source and target distribution. Therefore, in our work we adopt non-parametric statistical hypothesis testing framework to detect shift without knowing the actual parameters of the population.

SHIFT DETECTION IN NLP Within NLP, Arora, Huang, and He [5] focused on out-of-distribution texts and two approaches for detection. Shifts are categorized into *background* shift and *semantic* shift. Model calibration and density estimation are investigated for shift detection across 14 pairs of natural language understanding datasets. Comparing density estimation methods and calibration methods. We investigate different types of shifts than these works.

Given the importance of shift detection, a number of datasets have been developed [103, 137], however, they are not for the widely used task of NER.

Our work adds to this existing literature. First, we employ widely used labelled NER datasets and compare not only changes in fields (e.g. science to finance) but also changes in text style (e.g. news style text to social media style text). Second, we test the impact of representation choice on shift detection. Lastly, we provide new evidence for the performance impact of distribution shifts on task performance.

2.3 METHODOLOGY

Our methodology consists of the following steps: data collection; representation choice; statistical hypothesis testing and shift impact measurement. For data collection, we acquire datasets from different domains. Domains are characterized by their language usage arising from the style employed to the use of language particular to given field usage. For all datasets, both the space of input text and the space of labels are considered. In terms of representations, two types of representations are used for the input and one for labels. Statistical hypothesis testing appropriate for each representation is used to detect distribution shifts. The calculated statistics are then used to measure the extent of a shift. Lastly, the impact of each shift on model performance is ascertained. We now walk through each of these steps in detail.

2.3.1 *Data collection*

We collected 12 datasets from different domains covering news, social media, encyclopedic content, finance, science, emails, and business. Table 2.1 shows the list of datasets with the published year, document source, domains and the number of entity types. Table 2.3 shows the list of datasets and their entity types. We group the datasets into five categories, which we now describe in turn.

Wiki data

GUM [227] (the Georgetown University Multilayer Corpus) is collected and expanded as part of the curriculum of a course. The current corpus contains texts from public wikis (e.g. Wikinews, Wikivoyage, wikiHow, Wikipedia) as well as social media sites (e.g. Reddit, Youtube). Example types include *event*, *time*, *animal* and *abstract*. **Wikigold** [7] is a gold-standard NER dataset sourced from Wikipedia. Wikigold has standard types such as *person* and *organization*.

Informal text

Formal texts such as in news and Wikipedia are normally verified by multiple people sometimes even experts. Hence, the majority of text has correct grammar and spelling. In comparison, user-generated informal data such as social media texts, often contain less formal language usage characterized by slang, poor grammar, misspellings, the use of satire, etc. **BTC** [52](Broad Twitter Corpus) is a NER dataset where the source data is from Twitter that not only has tweets on general topics but also on specific topics such as disasters. BTC includes 3 types: *person*, *location* and *organization*. **WNUT17** [53] is a NER dataset where the text sources are Reddit, Twitter, YouTube and StackExchange comments. WNUT17

Corpus	Data Size	# Types	Entity Types
GUM	3,424	11	Organization, Person, Location, Event, Abstract, Object, Time, Substance, Plant, Quantity, Animal
Wikigold	1,688	4	Organization, Person, Location, Miscellaneous
BTC	9,318	3	Organization, Person, Location
W-NUT17	5,591	6	Organization, Person, Location, Group, Product, Creativework
CEREC	2,031	4	Organization, Person, Location, Digits
AnEM	4,423	11	Multi-tissue_structure, Organism_substance, Organism_subdivision, Organ, Cellular_component, Cell, Immaterial_anatomical_entity, Tissue, Pathological_formation, Anatomical_system, Developing_anatomical_structure
i2b2-06	40,280	7	Person, Location, ID, Date, Phone, Age
SEC-Filings	1,435	4	Organization, Person, Location, Miscellaneous
SciERC	2,687	6	Material, OtherScientificTerm, Generic, Method, Task, Metric
Re3d	948	10	Organization, Person, Location, Temporal, Nationality, Quantity, Weapon, Money, MilitaryPlatform, DocumentReference
CoNLL-03	17,350	4	Organization, Person, Location, Miscellaneous
OntoNotes	17,760	18	Organization, Location, Person, Work_of_Art, Cardinal, Event, NORP, Date, FAC, Quantity, Ordinal, Time, Product, Percent, Money, Law, Language

Table 2.3: List of NER datasets with corresponding entity types. Data size represents the number of sentences in each dataset.

focuses especially on emerging and rare entities. The dataset contains 6 types, including *creative*, *corporation* and *product* besides common types. **CEREC** [48] is a large-scale corpus for entity resolution in email conversations. The emails are taken from the first large public corpus the Enron Email Corpus [101] which contains emails of 150 employees of the Enron Corporation. Cerec contains standard types such as *person* and *digits* type.

Specific fields

AnEm [153] is a corpus annotated with species-independent anatomical entity mentions. The texts are from academic papers from the biomedical scientific literature. AnEm contains 11 domain-specific types such as *organ*, *cell* and *organism_substance*. **i2b2** [199] is a corpus that contains unstructured clinical notes from the Research Patient Data Registry at Partners Healthcare. The dataset consists of 8 types such as *hospital*, *phone* and *doctor*. **SEC-filings** [183] (U.S. Security and Exchange Commission filings) is a randomly selected and manually annotated finance dataset. The texts are from public-domain financial reports. The dataset includes standard types from CoNLL, i.e. *organization*, *person*, *location* and *misc*. **SciERC** [134] is a dataset that includes annotations for scientific entities in 500 scientific abstracts from AI conferences and workshop proceedings. The dataset focuses on scientific related types including *material*, *method* and *task*. **Re3d** [59] was constructed from documents that are relevant to the defence and security analysis domain, specifically, focusing on the topic of the conflict in Syria and Iraq. It includes domain-specific types such as *weapons* and *military platform*.

News

CoNLL-03 [196][1] is a dataset where the texts are taken from the Reuters news stories from 1996 to 1997. It contains the standard types including *person*, *location*, *organization* and *misc*.

General

OntoNotes [219][2] is a large annotated corpus that consists of various genres of texts including news, conversational telephone speech, weblogs, newsgroups, broadcast, and talk shows). OntoNotes include a large variety of types (18) including common types and less common ones such as *money* and *percent*.

Even though the datasets are grouped into five categories, there is still overlap. Wiki-based datasets and OntoNotes or CoNLL belong to different categories, but they might share many similar general entities. This is because

1 We use only the English data.
2 Similiar to CONLL, only English data is used.

common entities in the news are highly likely to have Wikipedia pages. Intuitively, the "similarity" between datasets in the wiki group should be larger. Conversely, the "similarity" between the domain-specific financial dataset SEC and the news dataset CoNLL should be smaller. We introduce methods to statistically quantify the distance between datasets in the following sections.

For all datasets, we preprocess them as follows. Duplicates are removed to prevent overfitting. Labels are unified across datasets shown in the Appendix a, Listing a.1. Different datasets use different labels to refer to the same type. Hence, to better compare performance, we unify the labels with the same semantic meanings. For example, 'person' and 'PER' will be unified under the same label.

2.3.2 *Shift detection and measurement*

We use statistical testing to determine and measure shifts between datasets. Formally, given a labeled source domain data $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\} \sim p$ and labeled target domain data $\{(\mathbf{x'}_1, \mathbf{y'}_1), ..., (\mathbf{x'}_n, \mathbf{y'}_n)\} \sim q$, shift detection determines whether p equals q. The null hypothesis is $H_0 : p = q$ and the alternative hypothesis $H_0 : p \neq q$. The statistical values are used as shift measurements. Both shifts occurring in the input distribution $p(\mathbf{x})$ and the label distribution $p(\mathbf{y})$ are investigated.

When forming the dataset pairs for the hypothesis testing, the distance functions we use are bi-directional, meaning that given a function Dist calculating distance, Dist(p, q) = Dist(q, p). Therefore, we measure the distribution shifts using combinations without repetition. Additionally, we include the distance of each dataset to itself as a sanity check. This approach results in 66 unique combinations plus 12 self-comparisons, in a total of 78 pairs.

We now discuss the representations we use for the datasets and the corresponding statistical tests we employ.

2.3.2.1 *Representation for input space*

We investigate two different representations for the input space.

Word frequencies: in this setting, $\mathbf{x}$ represents the frequency of each word. The underlying assumption is that the occurrences of words within a dataset indicate how important a word is. The word frequency distribution over the vocabulary represents the dataset.

Distributional representation: In this setting, each instance of $\mathbf{x}$ is an n-dimensional vector representation of a sentence within a dataset. Sentence-BERT [175] is used to encode each sentence. The idea behind sentence-BERT is that semantically similar sentences are closer in vector space [175]. The data points in this n-dimensional space are the distribution for each dataset.

2.3.2.2 *Representation for label space*

Within the NER task, datasets from different domains have different types of entities. We use category counts as our label distribution $p(\mathbf{y})$. Among different domains, the most general types include Person, Organization, Location and Miscellaneous. As mentioned in Section 2.3.1, we unify labels with the same semantic meanings. We note that very field-specific datasets will have a different label space than more general datasets.

Following recent work on distribution shifts, for label space, we formulate the problem as one of *unseen data shift* where some attribute values are unseen under p but are seen under q [221]. For example, the type *Method* might have zero observation in many datasets such as in CoNLL and Wikigold, but it will have many observations in dataset SciERC. However, it does not necessarily mean that there are no entities that have the type Methods in the Wikigold dataset, but due to specific data generation processes, those entities are not annotated. We see this as an outcome of different sampling processes. We assume all datasets share a common label set $\mathbf{A}^l$ and some labels in the set are unseen in p but are seen in q due to systematic sampling error.

2.3.2.3 *Statistical Hypothesis Testing*

For each type of representation, a different statistical test is necessary, which we now detail. Shift decisions are reported based on the significant level. By default, we use .05 as the significant threshold for all tests. Furthermore, we use this statistical testing as a means to measure distribution shift and draw a connection between shift and performance.

CHI-SQUARED TEST For frequency distributions of input and label count distribution, each sample x_n is one categorical value that represents word occurrence in the domain. We adopt Pearson's Chi-Squared test, a parametric test for determining if two frequency distributions are the same. The crucial underlying assumption is that a corpus is modelled as a sequence of independent Bernoulli trials. The relevant statistic χ^2 can be computed as:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{C} \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

where O_i is the observed value for category i and E_i is the expected value for category i. All word occurrences below 5 are filtered out.

There has been a long debate if the chi-squared test, or statistical testing in general, should be applied for corpus linguistics [71]. However, it is still widely employed within the literature [166]. Given that the distribution shift literature also employs chi-squared testing, we also make use of it here.

We employ two data processing procedures while using Chi-Squared tests. First, before applying the Chi-Squared test to data, we implement a normalization procedure to ensure that both the observed and expected values are on the same scale [198]. This normalization enhances the robustness of the test to different sample sizes. Second, by design, label distribution may contain a considerable number of zeros for certain categories. Since Chi-Squared test is not viable when dividing by zero, we added a small constant ($1e-5$) to each category to ensure that we obtain results without changing the numerical meaning of the results [3].

Another potential test for this sort of distribution is the Kolmogorov-Smirnov (KS) two-sample test. However, this test fits the cumulative distribution which requires values to be sorted. Sorting items in a vocabulary is not meaningful.

MAXIMUM MEAN DISCREPANCY (MMD) For multi-dimensional representations obtained from sentence-BERT, we employ MMD [70], a nonparametric kernel-based two-sample test to determine if two samples are drawn from two different distributions p and q. MMD tries to calculate the L_2 distance between the mean embeddings μ_p and μ_q of the distributions in a reproducing kernel Hilbert space $\mathcal{F}$ as:

$$\mathbf{MMD}^2(P, Q) = < \mu_p, \mu_p > -2 < \mu_p, \mu_q > + < \mu_q, \mu_q > .$$

Empirically, we use the unbiased estimate of the squared MMD statistic:

$$\begin{aligned}\mathbf{MMD}^2 = &\frac{1}{m(m-1)} \sum_{i \neq j}^{m} \kappa(x_i, x_j) \\ &+ \frac{1}{n(n-1)} \sum_{i \neq j}^{n} \kappa(x'_i, x'_j) \\ &- \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \kappa(x_i, x'_j).\end{aligned}$$

where the kernel is computed with a squared exponential function $\kappa(x, \tilde{x}) = e^{-\frac{1}{\sigma}|x-\tilde{x}|^2}$. σ is the median distance between points [70].

2.3.3 *Impact on Performance*

The last step in our method is to detect how the shift affects model performance. Our hypothesis is that *as the degree of distribution shift increases, so does*

3 This approach is inspired by the methods used to address the divide by zero problem in multi-class logistic regression in machine learning

the likelihood that a model makes an error and hence the degree of this error will also increase. As one of the most widely used pre-trained language models, we use BERT [54] to measure performance. Specifically, we measure the effect of shifts from $p(x)$ and shifts from $p(y)$ on performance.

We fine-tune the BERT model on one dataset and then test its performance across all datasets to identify any performance degradation. When adapting BERT for NER, we treat it as a sequence-to-sequence task utilizing the BIO tagging scheme where each token in a sentence is tagged with Begin, Inside, or Outside to indicate named entity boundaries. Then, we add one fully-connected dense layer for predicting tags. A cross-entropy loss is used for calculating the loss between predicted tag sequence and the golden tag sequence.

Specifically, we first split each dataset into a training set and an inference set, ensuring that all models trained on one dataset can also be tested on the same dataset. We follow the classic machine learning split ratio of 80:20, training on 80% of the dataset and testing on 20%. We then pair any two of the datasets and use the training set of the first as the source domain and the inference set for the second as the target domain. We fine-tune the original baseline model on the source training set and evaluate on the target inference set. Fine-tuning is performed for 10 epochs. Similar to the original BERT paper, we use a batch size of 32 and a learning rate of 5e-5. To ensure robustness in our results, we report the average performance across five trials, each with different random samplings. We train and test our model on GPU GeForce 1080Ti with 11GB GPU RAM. The fine-tuned BERT model's architecture consists of 12 layers of attention blocks, with each layer having a hidden size of 768 in the embedding layers. We follow the standard configuration of the BERT-base model, which comprises approximately 110 million parameters [54]. When we finetune it for NER task, we add one more fullyconnected dense layer. The number of parameters for each neuron is hidden size + bias (768 + 1) = 769. The final number of parameters added is $\mathtt{num\_tags} * 768 \approx 15{,}380$ parameters. Our code for both the hypothesis testing and evaluation is available as supplementary material.

To draw a connection between distribution shifts and performance degradation, we calculate the correlation between the measurement $\mathtt{shift}$ and the performance difference $\mathtt{perf}_{ab}$ between any two datasets a and b.

2.3.4 *Experimental Setup*

We conduct various experiments under different setups. For shift detection, we verify the validity of the tests on the sampled datasets of the same corpus. If the results indicate no shift detected, this implies that the testing has effectively validated that the two distributions are the same. We subsequently apply tests to all pairs of datasets.

As illustrated in Table 2.3, the datasets exhibit varying sizes. In general, ML models perform better when fine-tuned with more data. To mitigate the poten-

tial biases derived from data size, we uniformly sample a subset of 948 samples from each dataset and perform all experiments. This is because the minimum number of samples among all datasets is 948 from the Re3d dataset. In practice, engineers often fine-tune the model on the full dataset for better performance. To investigate the correlation under this scenario, we also employ identical tests and performance measures on the original full-sized datasets, the results of which are included in the Table 2.10, 2.9 and 2.11.

For the performance measures, BERT is pre-trained on a particular scope of texts and may favour datasets from certain domains. To address this potential bias, we utilize both BERT-base and BioBERT-base [109] and compare their respective performance outcomes.

2.4 RESULTS AND DISCUSSION

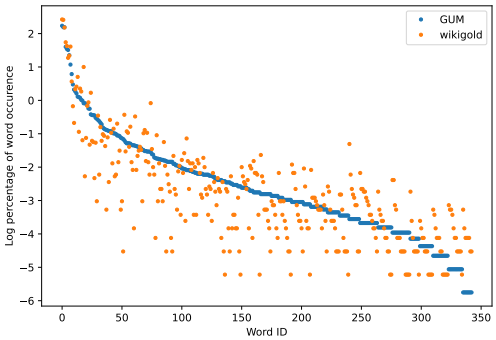

Figure 2.3: GUM and Wikigold.

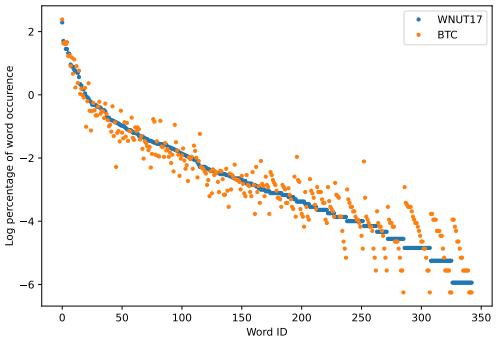

Figure 2.4: WNUT-17 and BTC.

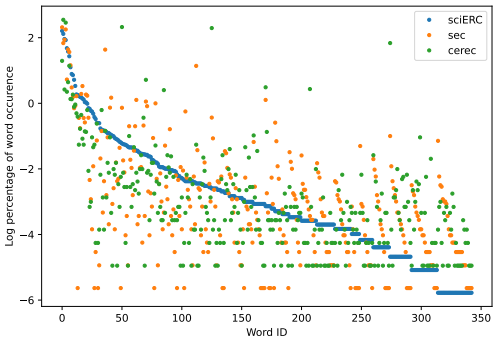

Figure 2.5: SciERC, SEC and CEREC.

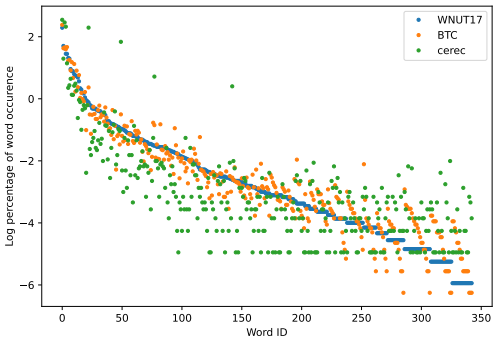

Figure 2.6: WNUT-17, BTC and CEREC.

We now present the results of applying the method detailed above. We begin with an analysis of the input datasets to verify our hypothesis about the shift between distributions representing shift between domains.

2.4.1 *Datasets analysis*

Figure 2.3, 2.4, 2.5, and 2.6 show the word frequency plots on selected pairs of datasets. WNUT-17 and BTC both include text from Twitter, and we see that word frequency is similar across both datasets. Conversely, in the case of SciERC, SEC and Cerec datasets, which more distinctly represent different domains, we observe greater dispersion within their respective word frequency distributions. Intuitively, these results suggest that word frequency distributions can serve as an indicator of a domain.

2.4.2 *Hypothesis testing*

2.4.2.1 *Chi-squared testing for input distribution*

Table 2.4 shows the chi-squared testing on both sampled dataset pairs and original-sized dataset pairs. For Chi-squared tests, the shift is made based on the p-value of the testing. A common cutoff for rejecting the null hypothesis in this context is a p-value less than 0.05, indicating a statistically significant difference in distributions. The results of the Chi-squared tests are provided as a proximity for the distribution differences. The higher the value, the greater the disparity. The distance between the same distribution is also reported as a sanity check. When the source and target distributions are equivalent, the testing indicates that no shift is detected. This indicates that the testing is capable of identifying when two distributions are identical.

For the full-sized datasets, Table 2.4 reveals that among the 78 dataset pairs, 13 pairs are detected with shifts. Meanwhile, Table 2.5 indicates that for the sampled datasets, out of the same 78 pairs, 22 pairs are detected to have shifts. These results suggest that the test is more sensitive to identifying shifts when there is a smaller sample size. On closer inspection of the dataset pairs, we observe that out of the 13 shift-detected full-sized pairs, 11 pairs are also detected in the sampled datasets, which reaches an approximately 84.6% agreement. Additionally, the results presented in Table 2.4 reveal that a higher Chi-Squared value does not necessarily imply the detection of a shift. For instance, while the OntoNotes and i2b2 datasets have high Chi-Squared values, no shift is detected. This outcome could arise due to the data samples being non-representative of the full distribution, thereby resulting in the test's inability to make a confident conclusion.

Analyzing these results, we note that BTC and WNUT-17 datasets have the smallest distance, which is inline with the frequency plots in Figure 2.4. On the other hand, the BTC dataset and SEC finance dataset have the furthest distance, which, as expected, reflects that these two datasets have very different text styles. One surprising outcome is GUM and SciERC which have a relatively small distance using this representation while being from what appear

to be different domains. These examples illustrate that this test can quantify the distance between datasets.

Table 2.4: Input chi-squared statistics for all combinations without repetition of datasets. The table is ordered by the chi-squared value following ascending order. All word occurrences that are below 5 are filtered for the effective usage of chi-squared testing. The symbol ♣ indicates there are shifts detected.

Source data	Target data	Statistics	Shift decision
conll	conll	0.00	
cerec	cerec	0.00	
ontonotes	ontonotes	0.00	
i2b2	i2b2	0.00	
GUM	GUM	0.00	
AnEM	AnEM	0.00	
BTC	BTC	0.00	
WNUT17	WNUT17	0.00	
wikigold	wikigold	0.00	
re3d	re3d	0.00	
sec	sec	0.00	
sciERC	sciERC	0.00	
BTC	WNUT17	96.04	
GUM	WNUT17	123.53	
GUM	BTC	127.48	
ontonotes	WNUT17	144.48	
ontonotes	re3d	149.69	
GUM	wikigold	191.42	
wikigold	re3d	232.63	
GUM	re3d	257.93	
ontonotes	GUM	292.86	
AnEM	re3d	302.56	
AnEM	wikigold	337.45	
ontonotes	wikigold	346.67	
GUM	sciERC	349.31	
GUM	sec	385.12	
AnEM	WNUT17	401.39	
i2b2	re3d	426.14	
re3d	sec	471.70	
ontonotes	BTC	479.08	
AnEM	BTC	500.50	
wikigold	sciERC	544.95	
AnEM	sec	548.65	
i2b2	GUM	584.29	
wikigold	sec	642.16	
BTC	wikigold	787.59	
re3d	sciERC	789.35	
WNUT17	wikigold	794.37	
ontonotes	AnEM	820.95	
i2b2	AnEM	823.57	
ontonotes	sec	853.52	
AnEM	sciERC	913.34	
GUM	AnEM	961.50	
conll	WNUT17	1,047.27	
i2b2	BTC	1,086.96	
conll	cerec	1,089.80	
cerec	re3d	1,148.76	
i2b2	wikigold	1,164.45	
sec	sciERC	1,171.78	

Continued on next page

Source data	Target data	Statistics (cont.)	Shift decision
cerec	WNUT17	1,378.34	
WNUT17	sciERC	1,434.51	
WNUT17	re3d	1,478.79	
BTC	sciERC	1,510.25	
conll	BTC	1,524.60	
WNUT17	sec	1,656.90	
cerec	wikigold	1,727.73	♣
ontonotes	sciERC	1,728.98	
cerec	AnEM	1,757.95	♣
i2b2	sec	1,777.99	
cerec	BTC	1,879.36	
conll	sec	2,027.94	
conll	ontonotes	2,092.56	
conll	wikigold	2,100.30	
cerec	sec	2,240.83	♣
conll	i2b2	2,257.31	
i2b2	WNUT17	2,653.34	
i2b2	sciERC	2,808.05	♣
conll	GUM	2,877.06	
conll	re3d	3,124.31	♣
ontonotes	i2b2	3,838.24	
cerec	GUM	4,310.95	♣
BTC	re3d	4,513.72	♣
BTC	sec	4,568.51	♣
cerec	ontonotes	4,726.86	♣
conll	sciERC	6,930.89	♣
cerec	sciERC	7,169.87	♣
conll	AnEM	7,186.44	♣
cerec	i2b2	7,927.04	♣

Table 2.5: Input chi-squared statistics for sampled datasets. The symbol ♣ indicates a detected shift.

Source data	Target data	Statistics	Shift decision
conll	conll	0.00	
cerec	cerec	0.00	
ontonotes	ontonotes	0.00	
i2b2	i2b2	0.00	
GUM	GUM	0.00	
AnEM	AnEM	0.00	
BTC	BTC	0.00	
WNUT17	WNUT17	0.00	
wikigold	wikigold	0.00	
re3d	re3d	0.00	
sec	sec	0.00	
sciERC	sciERC	0.00	
GUM	BTC	75.67	
GUM	WNUT17	81.64	
BTC	WNUT17	93.89	
ontonotes	WNUT17	98.60	
ontonotes	BTC	124.59	
ontonotes	re3d	131.39	
AnEM	BTC	158.86	
GUM	wikigold	162.41	
i2b2	wikigold	172.97	
AnEM	WNUT17	189.54	

Continued on next page

Source data	Target data	Statistics (cont.)	Shift decision
wikigold	sciERC	191.47	
AnEM	re3d	195.15	
GUM	re3d	207.65	
i2b2	GUM	211.51	
wikigold	re3d	213.81	
GUM	AnEM	214.20	
GUM	sciERC	244.28	
ontonotes	wikigold	249.68	
i2b2	AnEM	258.37	
conll	AnEM	258.89	
ontonotes	AnEM	283.88	
AnEM	wikigold	292.75	
conll	BTC	295.26	
i2b2	BTC	296.28	
re3d	sec	316.50	
i2b2	re3d	320.14	
wikigold	sec	327.45	
ontonotes	GUM	345.13	
ontonotes	i2b2	357.75	
GUM	sec	401.75	
re3d	sciERC	404.84	
WNUT17	re3d	407.13	
AnEM	sec	433.05	
conll	re3d	433.08	
i2b2	WNUT17	490.46	
sec	sciERC	501.09	
cerec	WNUT17	603.98	
AnEM	sciERC	635.92	
ontonotes	sec	716.12	
BTC	wikigold	748.72	
conll	sec	760.04	
conll	WNUT17	778.92	
cerec	re3d	815.43	♣
conll	GUM	832.89	
cerec	wikigold	836.13	♣
WNUT17	wikigold	861.27	
cerec	i2b2	875.33	♣
i2b2	sec	881.32	♣
cerec	AnEM	892.46	♣
i2b2	sciERC	905.32	♣
cerec	GUM	1,89.98	♣
conll	i2b2	1,425.79	♣
cerec	sec	1,428.67	♣
BTC	sciERC	1,643.86	♣
cerec	BTC	1,795.81	♣
conll	cerec	1,810.52	♣
cerec	ontonotes	1,961.29	♣
WNUT17	sciERC	2,16.53	♣
conll	wikigold	2,145.54	♣
conll	ontonotes	2,348.69	♣
BTC	re3d	2,480.40	♣
WNUT17	sec	2,590.02	♣
ontonotes	sciERC	2,608.52	♣
conll	sciERC	3,51.15	♣
BTC	sec	4,274.04	♣
cerec	sciERC	4,769.50	♣

2.4.2.2 *MMD testing for input distribution*

For the distributional representations, we apply MMD with a different number of samples (i.e. embedded sentences) from n = [5, 50, 200, 500, 1000, 2000]. Table 2.6 shows the results of these tests. The table is ordered by the scores generated using 2000 samples. We measure the difference between a dataset and itself as a sanity check.

Intuitively, the MMD test evaluates whether there is a significant difference between two distributions: a higher MMD value suggests a greater disparity between the distributions. The test essentially disproves the null hypothesis, that the distributions are identical, when the MMD statistic is significantly high. In our study, the MMD values can appear negative due to estimation errors in smaller samples or due to the kernel choice affecting the calculation. However, the absolute value of MMD should be considered. Typically, a threshold for significance is set, above which the null hypothesis can be rejected. We use 0.05 as our threshold.

CoNLL, a widely used benchmark dataset in NER, is surprisingly far from other datasets in the distance measured by the chi-squared test. However, with MMD tests, the distance is fairly close. This is an indication that sentence-level representation provides more information than word-frequency representation.

Table 2.6: MMD statistics for **input embedding distribution** on full-sized datasets with a different number of samples. This is ordered by the distance between pairs of datasets with 2,000 samples. Sign ⋆ indicates there is a shift detected.

Source data	Target data	Number of samples from test					
		5	50	200	500	1,000	2,000
AnEM	AnEM	-0.2542	-0.0281	-0.0073	-0.0029	-0.0015	-0.0007
BTC	BTC	-0.2906	-0.0286	-0.0070	-0.0028	-0.0014	-0.0007
GUM	GUM	-0.2521	-0.0277	-0.0071	-0.0029	-0.0015	-0.0007
WNUT17	WNUT17	-0.2521	-0.0289	-0.0073	-0.0030	-0.0015	-0.0007
cerec	cerec	-0.2776	-0.0286	-0.0072	-0.0029	-0.0014	-0.0007
conll	conll	-0.2653	-0.0284	-0.0073	-0.0029	-0.0014	-0.0007
i2b2	i2b2	-0.2520	-0.0284	-0.0071	-0.0028	-0.0014	-0.0007
ontonotes	ontonotes	-0.2587	-0.0254	-0.0067	-0.0029	-0.0014	-0.0007
sciERC	sciERC	-0.2890	-0.0284	-0.0071	-0.0028	-0.0014	-0.0007
wikigold	wikigold	-0.2695	-0.0282	-0.0072	-0.0029	-0.0015	-0.0008
sec	sec	-0.2831	-0.0273	-0.0068	-0.0027	-0.0014	-0.0009
re3d	re3d	-0.2600	-0.0279	-0.0071	-0.0029	-0.0015	-0.0015
GUM	WNUT17	0.6140⋆	0.1566⋆	0.0964⋆	0.0619⋆	0.0425	0.0271
GUM	wikigold	0.4367⋆	0.2047⋆	0.1266⋆	0.0828⋆	0.0615⋆	0.0294
BTC	WNUT17	0.2581⋆	0.0503⋆	0.0509⋆	0.0466	0.0353	0.0311
conll	wikigold	0.2646⋆	0.1069⋆	0.0669⋆	0.0385	0.0479	0.0348
WNUT17	wikigold	0.4361⋆	0.0789⋆	0.0532⋆	0.0297	0.0283	0.0406
GUM	AnEM	0.4231⋆	0.1762⋆	0.0975⋆	0.0727⋆	0.0607⋆	0.0411
conll	WNUT17	0.4148⋆	0.0678⋆	0.0402	0.0338	0.0532⋆	0.0443
conll	GUM	0.3968⋆	0.1629⋆	0.1158⋆	0.0870⋆	0.0934⋆	0.0450
i2b2	AnEM	0.2739⋆	0.0860⋆	0.0557⋆	0.0535⋆	0.0461	0.0477
AnEM	wikigold	0.2688⋆	0.1235⋆	0.0717⋆	0.0520⋆	0.0503⋆	0.0478
ontonotes	re3d	0.2496⋆	0.2194⋆	0.1538⋆	0.0500	0.0535⋆	0.0480

Continued on next page

Source data	Target data	Number of samples from test (cont.)					
		5	50	200	500	1,000	2,000
cerec	WNUT17	0.3337*	0.0627*	0.0509*	0.0467	0.0465	0.0484
ontonotes	GUM	0.5314*	0.3557*	0.2142*	0.1115*	0.0890*	0.0530*
ontonotes	WNUT17	0.4052*	0.2008*	0.1389*	0.0457	0.0423	0.0534*
cerec	GUM	0.3492*	0.1526*	0.1075*	0.0874*	0.0824*	0.0552*
ontonotes	BTC	0.2600*	0.2338*	0.1792*	0.0792*	0.0686*	0.0572*
conll	cerec	0.1973*	0.0861*	0.0626*	0.0508*	0.0738*	0.0574*
GUM	BTC	0.3451*	0.1934*	0.1547*	0.1223*	0.0994*	0.0582*
cerec	wikigold	0.2564*	0.1174*	0.0783*	0.0634*	0.0615*	0.0605*
conll	AnEM	0.2566*	0.1009*	0.0656*	0.0606*	0.0812*	0.0607*
AnEM	WNUT17	0.4262*	0.1042*	0.0554*	0.0449	0.0487	0.0617*
conll	BTC	0.1510*	0.0878*	0.0817*	0.0761*	0.0871*	0.0632*
ontonotes	wikigold	0.3596*	0.1709*	0.1677*	0.0701*	0.0650*	0.0666*
BTC	wikigold	0.2147*	0.1211*	0.1080*	0.0824*	0.0687*	0.0682*
i2b2	wikigold	0.3163*	0.1125*	0.0753*	0.0686*	0.0642*	0.0683*
conll	ontonotes	0.3720*	0.2402*	0.1244*	0.0697*	0.0970*	0.0686*
GUM	re3d	0.4959*	0.2415*	0.1646*	0.1108*	0.1087*	0.0728*
i2b2	GUM	0.4385*	0.1880*	0.1217*	0.1121*	0.0940*	0.0735*
conll	i2b2	0.2461*	0.0980*	0.0729*	0.0724*	0.0836*	0.0737*
conll	re3d	0.3366*	0.1395*	0.0453	0.0541*	0.0952*	0.0753*
cerec	BTC	0.0892*	0.0911*	0.0928*	0.0907*	0.0865*	0.0763*
GUM	sciERC	0.3731*	0.1781*	0.1375*	0.1213*	0.1008*	0.0778*
WNUT17	re3d	0.3548*	0.0967*	0.0740*	0.0483	0.0650*	0.0784*
wikigold	re3d	0.3274*	0.1211*	0.0898*	0.0638*	0.0760*	0.0793*
i2b2	WNUT17	0.4524*	0.0822*	0.0651*	0.0710*	0.0656*	0.0797*
BTC	re3d	0.2138*	0.1317*	0.1071*	0.0878*	0.0910*	0.0807*
cerec	AnEM	0.2287*	0.1215*	0.0747*	0.0754*	0.0846*	0.0815*
ontonotes	AnEM	0.3626*	0.2827*	0.1698*	0.0845*	0.0846*	0.0845*
cerec	i2b2	0.1851*	0.0734*	0.0571*	0.0755*	0.0769*	0.0858*
cerec	ontonotes	0.3310*	0.2643*	0.1853*	0.0952*	0.0933*	0.0865*
wikigold	sciERC	0.2372*	0.1347*	0.1095*	0.1055*	0.0981*	0.0918*
AnEM	BTC	0.1894*	0.1244*	0.1082*	0.1025*	0.1010*	0.0948*
WNUT17	sciERC	0.3590*	0.0939*	0.0878*	0.0880*	0.0867*	0.0964*
AnEM	re3d	0.3190*	0.1483*	0.0982*	0.0819*	0.0999*	0.0994*
cerec	re3d	0.2920*	0.1363*	0.1039*	0.0878*	0.1045*	0.1008*
AnEM	sciERC	0.2466*	0.1329*	0.0922*	0.1032*	0.1069*	0.1016*
conll	sciERC	0.2141*	0.1163*	0.1087*	0.1114*	0.1292*	0.1065*
cerec	sec	0.1203*	0.1248*	0.1239*	0.1231*	0.1081*	0.1091*
i2b2	BTC	0.2184*	0.1072*	0.1113*	0.1160*	0.1094*	0.1102*
cerec	sciERC	0.1908*	0.1145*	0.1050*	0.1144*	0.1175*	0.1116*
ontonotes	i2b2	0.3943*	0.2750*	0.1961*	0.1162*	0.1077*	0.1119*
wikigold	sec	0.2417*	0.1701*	0.1441*	0.1281*	0.1045*	0.1135*
GUM	sec	0.3692*	0.2313*	0.1893*	0.1633*	0.1321*	0.1144*
conll	sec	0.1941*	0.1385*	0.1251*	0.1226*	0.1294*	0.1167*
ontonotes	sciERC	0.3224*	0.2789*	0.2021*	0.1281*	0.1205*	0.1173*
AnEM	sec	0.2065*	0.1559*	0.1331*	0.1321*	0.1207*	0.1277*
i2b2	re3d	0.3472*	0.1175*	0.1160*	0.1131*	0.1236*	0.1286*
WNUT17	sec	0.3779*	0.1311*	0.1277*	0.1186*	0.1059*	0.1289*
BTC	sciERC	0.1668*	0.1354*	0.1548*	0.1544*	0.1449*	0.1342*
re3d	sciERC	0.2966*	0.1667*	0.1414*	0.1280*	0.1416*	0.1374*
i2b2	sciERC	0.2644*	0.1371*	0.1261*	0.1480*	0.1416*	0.1435*
ontonotes	sec	0.3163*	0.3097*	0.2311*	0.1569*	0.1379*	0.1438*
i2b2	sec	0.2510*	0.1378*	0.1387*	0.1521*	0.1336*	0.1462*
re3d	sec	0.2838*	0.1956*	0.1589*	0.1482*	0.1455*	0.1533*
sec	sciERC	0.1811*	0.1640*	0.1612*	0.1690*	0.1497*	0.1536*

2.4.2.3 *Label distribution*

To detect category shift in label distribution, we utilized the Chi-squared test, as detailed in Section 2.3.4. This testing was performed on both the sampled and full-sized datasets, and the results are presented in Table 2.7 and Table 2.8. Results reveal a significant difference between the datasets that share the same categories and those that have different categories.

Datasets that are focused on specialized fields typically contain more specific labels. Consequently, the dissimilarity between these datasets and those from general domains is greater. For example, while the input shift between BTC and WNUT17 may be small, the label shift is relatively significant due to their distinct label spaces. For the NER task, generalizing model performance to datasets that have distinct categories is more challenging, as evidenced in the following section.

Table 2.7: Label distribution chi-squared testing statistics for all combinations without repetition of datasets. The table is ordered by the test value in ascending order.

Source data	Target data	Statistics	Shift Decision
conll	conll	0.00	
cerec	cerec	0.00	
ontonotes	ontonotes	0.00	
i2b2-06	i2b2-06	0.00	
GUM	GUM	0.00	
AnEM	AnEM	0.00	
BTC	BTC	0.00	
WNUT17	WNUT17	0.00	
wikigold	wikigold	0.00	
re3d	re3d	0.00	
sec	sec	0.00	
sciERC	sciERC	0.00	
conll	wikigold	0.04	
BTC	sec	0.07	
BTC	wikigold	0.51	
BTC	WNUT17	0.84	
BTC	re3d	1.22	
conll	sec	2.03	
wikigold	sec	4.24	
cerec	sec	326,074.25	♣
cerec	re3d	745,315.20	♣
cerec	wikigold	781,788.25	♣
cerec	WNUT17	841,992.77	♣
re3d	sec	1,310,152.93	♣
cerec	GUM	2,564,429.92	♣
cerec	BTC	3,257,665.19	♣
WNUT17	sec	5,112,738.62	♣
ontonotes	sec	5,223,820.29	♣
conll	cerec	5,780,109.23	♣
conll	re3d	6,382,202.06	♣
conll	WNUT17	7,210,082.54	♣
WNUT17	re3d	11,686,258.40	♣
ontonotes	re3d	11,940,158.70	♣

Continued on next page

Source data	Target data	Statistics (cont.)	Shift Decision
WNUT17	wikigold	12,258,168.10	♣
GUM	sec	12,466,422.81	♣
ontonotes	wikigold	12,524,494.36	♣
ontonotes	WNUT17	13,489,000.18	♣
wikigold	re3d	13,583,255.59	♣
conll	GUM	21,959,515.63	♣
AnEM	sec	22,315,222.59	♣
cerec	ontonotes	22,872,211.25	♣
conll	BTC	27,895,788.79	♣
GUM	re3d	28,494,675.69	♣
i2b2-06	sec	29,859,890.76	♣
GUM	wikigold	29,889,167.66	♣
GUM	WNUT17	32,190,918.71	♣
ontonotes	GUM	41,083,014.26	♣
ontonotes	AnEM	43,451,712.45	♣
GUM	AnEM	50,349,808.09	♣
AnEM	re3d	51,006,213.20	♣
cerec	i2b2-06	51,746,456.14	♣
ontonotes	BTC	52,188,909.35	♣
AnEM	wikigold	53,502,390.51	♣
AnEM	WNUT17	57,622,585.01	♣
i2b2-06	re3d	68,251,165.27	♣
i2b2-06	wikigold	71,591,287.74	♣
ontonotes	i2b2-06	74,740,292.02	♣
i2b2-06	WNUT17	77,104,499.80	♣
conll	AnEM	84,469,257.53	♣
i2b2-06	AnEM	86,817,785.05	♣
GUM	BTC	124,546,587.29	♣
ontonotes	sciERC	160,417,887.32	♣
AnEM	sciERC	174,040,643.86	♣
GUM	sciERC	185,884,729.65	♣
cerec	AnEM	189,564,270.53	♣
conll	ontonotes	195,857,558.82	♣
conll	i2b2-06	197,142,187.31	♣
AnEM	BTC	222,941,642.42	♣
i2b2-06	GUM	234,834,691.92	♣
WNUT17	sciERC	279,748,960.06	♣
wikigold	sciERC	293,835,211.87	♣
re3d	sciERC	294,915,993.80	♣
i2b2-06	BTC	298,317,122.09	♣
conll	sciERC	311,849,153.00	♣
i2b2-06	sciERC	320,519,601.10	♣
BTC	sciERC	459,742,109.25	♣
sec	sciERC	555,622,639.19	♣

Table 2.8: Label distribution chi-squared testing statistics for sampled datasets.

Source data	Target data	Statistics	Shift Decision
conll	conll	0.00	
cerec	cerec	0.00	
ontonotes	ontonotes	0.00	
i2b2-06	i2b2-06	0.00	
GUM	GUM	0.00	
AnEM	AnEM	0.00	
BTC	BTC	0.00	
WNUT17	WNUT17	0.00	
wikigold	wikigold	0.00	

Continued on next page

Source data	Target data	Statistics (cont.)	Shift Decision
re3d	re3d	0.00	
sec	sec	0.00	
sciERC	sciERC	0.00	
conll	wikigold	0.04	
BTC	sec	0.10	
BTC	wikigold	0.52	
BTC	WNUT17	0.84	
BTC	re3d	1.11	
conll	sec	1.47	
wikigold	sec	2.60	
cerec	sec	334,179.62	♣
cerec	wikigold	839,683.36	♣
cerec	re3d	843,668.88	♣
cerec	WNUT17	917,624.79	♣
re3d	sec	1,066,254.19	♣
cerec	GUM	2,900,540.31	♣
cerec	BTC	3,687,180.00	♣
WNUT17	sec	4,377,314.31	♣
ontonotes	sec	4,780,033.07	♣
conll	cerec	5,836,613.50	♣
conll	re3d	6,491,937.00	♣
conll	WNUT17	7,061,035.89	♣
WNUT17	wikigold	10,998,734.44	♣
WNUT17	re3d	11,050,923.20	♣
GUM	sec	11,530,780.73	♣
wikigold	re3d	11,602,120.49	♣
ontonotes	wikigold	12,010,632.37	♣
ontonotes	re3d	12,067,622.20	♣
ontonotes	WNUT17	13,125,498.34	♣
AnEM	sec	20,244,050.26	♣
conll	GUM	22,319,385.77	♣
cerec	ontonotes	26,018,437.52	♣
i2b2-06	sec	26,465,192.08	♣
conll	BTC	28,372,520.38	♣
GUM	wikigold	28,973,014.22	♣
GUM	re3d	29,110,489.58	♣
GUM	WNUT17	31,662,383.01	♣
ontonotes	GUM	41,488,682.25	♣
ontonotes	AnEM	44,092,822.88	♣
AnEM	wikigold	50,866,559.82	♣
GUM	AnEM	50,979,958.03	♣
AnEM	re3d	51,107,918.60	♣
ontonotes	BTC	52,740,634.40	♣
AnEM	WNUT17	55,588,158.29	♣
cerec	i2b2-06	59,943,853.08	♣
i2b2-06	wikigold	66,498,217.78	♣
i2b2-06	re3d	66,813,748.00	♣
i2b2-06	WNUT17	72,670,797.12	♣
ontonotes	i2b2-06	74,879,460.31	♣
conll	AnEM	85,589,611.44	♣
i2b2-06	AnEM	87,686,295.03	♣
GUM	BTC	127,225,200.96	♣
ontonotes	sciERC	158,825,547.52	♣
AnEM	sciERC	173,009,643.23	♣
GUM	sciERC	183,633,507.68	♣
cerec	AnEM	187,632,540.01	♣
conll	i2b2-06	195,573,212.31	♣
conll	ontonotes	200,209,544.04	♣
AnEM	BTC	223,363,307.19	♣
i2b2-06	GUM	229,706,748.40	♣

Continued on next page

Source data	Target data	Statistics (cont.)	Shift Decision
WNUT17	sciERC	279,425,297.10	♣
i2b2-06	BTC	292,004,448.24	♣
wikigold	sciERC	294,289,564.04	♣
re3d	sciERC	305,681,552.91	♣
conll	sciERC	308,299,988.60	♣
i2b2-06	sciERC	315,852,394.93	♣
BTC	sciERC	451,014,301.26	♣
sec	sciERC	546,583,192.69	♣

2.4.3 *Performance measurement*

As noted in Section 2.3.4, we conducted four sets of experiments, including fine-tuning models on both original-sized and sampled datasets with 948 samples using both BERT-base and BioBERT-base models. All performance results are reported using the average of five trials with different random sampling.

Table 2.9, 2.10, and2.11 present the micro-averaged F1 performance of the models trained and tested on the sampled datasets with BERT, full-sized datasets with BERT, and full-sized datasets with BioBERT, respectively. The counterpart, which shows the micro-averaged F1 performance of the models trained and tested with sampled datasets with BioBERT, is shown in Appendix a, in Table a.1.

Each row in the table indicates the dataset the model is fine-tuned on. Correspondingly, the columns indicate the dataset which the fine-tuned model is tested upon. All performance results reported here are the average of 5 trials. The standard deviation of these results ranges from $0.0 \sim 0.07$. The last column reports the average F1 scores of all the test results, which represents the generalization ability of the model when fine-tuned on a specific dataset. The rows are ordered by this average F1 score.

Table 2.10 reveals that even though BTC and WNUT17 contain texts from the same domain, the model's generalization ability on WNUT17 decreases significantly when fine-tuned on BTC, as there is a significant label category shift between these two datasets.

Comparing Table 2.9 and Table 2.10, we observe that when we control the number of data samples, the average F1 scores tend to decrease. However, the overall rankings of the datasets are similar between the two tables, except for WNUT17 and Re3d. Using a subset of the original dataset reduces the generalization ability significantly, indicating the additional data samples in the original data set help improve the generalization. Conversely, for dataset Re3d, the generalization ability increases while the number of samples decreases, indicating that the additional data samples in the dataset harm the generalization.

Due to mutually exclusive sets of categories, we encounter many zero F1 scores on AnEM dataset and SciERC dataset. Even though fine-tuning helps improve the performance on the same dataset, the generalization ability is low, indicating that category shift has a significant impact on the performance.

Comparing Table 2.10 and Table 2.11, we observe that the fine-tuning performance is slightly impacted by the text on which these language models were pre-trained. However, the average F1 score rankings remain the same.

Table 2.9: Micro-average F1 score when the model is fine-tuned on the source dataset (the row) and tested on the target dataset (the column). Fine-tuning uses the BERT-base-uncased model. All performances are averaged over five trials. All datasets are sampled with 948 samples.

	conll	wikigold	cerec	BTC	re3d	i2b2-06	SEC	ontonotes	GUM	WNUT17	sciERC	AnEM	average f1
conll	**0.61**	0.38	0.24	0.28	0.21	0.13	0.13	0.23	0.06	0.24	0.00	0.00	**0.21**
wikigold	0.50	**0.51**	0.23	0.22	0.24	0.15	0.16	0.23	0.10	0.14	0.00	0.00	0.21
cerec	0.29	0.21	**0.74**	0.12	0.11	0.16	0.11	0.09	0.17	0.15	0.00	0.00	0.18
BTC	0.29	0.16	0.19	**0.62**	0.14	0.14	0.08	0.11	0.05	0.16	0.00	0.00	0.16
re3d	0.16	0.18	0.12	0.13	**0.47**	0.03	0.07	0.11	0.10	0.11	0.00	0.00	0.12
i2b2-06	0.09	0.09	0.12	0.17	0.01	**0.73**	0.02	0.02	0.01	0.07	0.00	0.00	0.11
SEC	0.09	0.05	0.06	0.06	0.02	0.00	**0.85**	0.08	0.01	0.05	0.00	0.00	0.11
ontonotes	0.14	0.08	0.03	0.05	0.11	0.02	0.04	**0.31**	0.03	0.05	0.00	0.00	0.07
GUM	0.08	0.09	0.12	0.05	0.08	0.01	0.03	0.03	**0.24**	0.04	0.00	0.00	0.06
WNUT17	0.08	0.07	0.11	0.07	0.02	0.08	0.01	0.02	0.02	**0.15**	0.00	0.00	0.05
sciERC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.24**	0.00	0.02
AnEM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.21**	0.02

Table 2.10: F1 scores on full-sized datasets. Fine-tuning uses the BERT-base model.

	conll	cerec	ontonotes	i2b2-06	wikigold	WNUT17	GUM	re3d	SEC	BTC	sciERC	AnEM	average f1
conll	**0.90**	0.35	0.31	0.13	0.62	0.26	0.12	0.33	0.10	0.31	0.00	0.00	**0.29**
cerec	0.35	**0.90**	0.13	0.10	0.31	0.14	0.21	0.19	0.08	0.20	0.00	0.00	0.25
ontonotes	0.28	0.14	**0.93**	0.05	0.26	0.14	0.08	0.17	0.06	0.17	0.00	0.00	0.23
i2b2-06	0.27	0.20	0.12	**0.99**	0.33	0.06	0.05	0.22	0.02	0.16	0.00	0.00	0.22
wikigold	0.57	0.25	0.30	0.10	**0.88**	0.13	0.13	0.30	0.13	0.20	0.00	0.00	0.21
WNUT17	0.43	0.32	0.29	0.15	0.44	**0.70**	0.10	0.24	0.09	0.27	0.00	0.00	0.21
GUM	0.20	0.17	0.05	0.01	0.23	0.04	**0.68**	0.28	0.04	0.06	0.00	0.00	0.21
re3d	0.19	0.25	0.15	0.04	0.19	0.11	0.14	**0.61**	0.06	0.20	0.00	0.00	0.21
SEC	0.22	0.14	0.19	0.05	0.25	0.09	0.05	0.12	**0.93**	0.21	0.00	0.00	0.21
BTC	0.59	0.35	0.32	0.14	0.62	0.24	0.14	0.34	0.15	**0.87**	0.00	0.00	0.20
sciERC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.57**	0.00	0.19
AnEM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.79**	0.19

2.4.4 *Correlation*

To investigate further how shifts impact model performance, we report the correlation between the testing statistics and performance differences in Figure 2.7,2.8, and 2.9. Within each plot, the x-axis represents the distances calculated by different hypothesis testing, and the y-axis represents the performance differences between a source dataset and a target dataset. The higher the performance distance is, the worse the generalization ability the model exhibits. Assuming there are datasets D_a and D_b. $\mathtt{Perf}_{ab}$ indicates the performance difference on D_a and D_b when the model is fine-tuned on source dataset D_s where

Table 2.11: F1 scores on full-sized datasets. Fine-tuning uses the BioBERT-base model.

	conll	cerec	ontonotes	i2b2-06	wikigold	WNUT17	GUM	re3d	SEC	BTC	sciERC	AnEM	average f1
conll	**0.88**	0.32	0.26	0.11	0.58	0.21	0.10	0.29	0.36	0.23	0.00	0.00	**0.28**
cerec	0.30	**0.87**	0.11	0.06	0.30	0.12	0.20	0.20	0.31	0.15	0.00	0.00	0.25
ontonotes	0.23	0.14	**0.92**	0.10	0.22	0.12	0.07	0.15	0.07	0.15	0.00	0.00	0.23
i2b2-06	0.33	0.22	0.17	**1.00**	0.36	0.09	0.09	0.26	0.23	0.16	0.00	0.00	0.23
wikigold	0.49	0.25	0.29	0.10	**0.84**	0.11	0.11	0.28	0.38	0.15	0.00	0.00	0.21
WNUT17	0.40	0.24	0.25	0.14	0.43	**0.61**	0.09	0.23	0.19	0.25	0.00	0.00	0.21
GUM	0.15	0.16	0.04	0.03	0.21	0.03	**0.63**	0.26	0.04	0.06	0.00	0.00	0.21
re3d	0.15	0.21	0.11	0.03	0.16	0.07	0.13	**0.53**	0.09	0.11	0.00	0.00	0.20
SEC	0.21	0.19	0.15	0.11	0.18	0.08	0.05	0.10	**0.90**	0.21	0.00	0.00	0.20
BTC	0.50	0.33	0.32	0.14	0.58	0.22	0.12	0.30	0.31	**0.85**	0.00	0.00	0.20
sciERC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.55**	0.00	0.19
AnEM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.77**	0.19

$s \in \{D_i \mid i = \{1, 2, ..., 12\}\}$. $Shift_{ab}$ is the distance between D_a and D_b with regarding to each statistical test. The correlation is calculated between $perf$ and $shift$.

Based on the presented plots, it is evident that the label category shift shows the most statistically significant correlation ($P < .0001$) with model performance. This finding suggests that category shift can serve as a reliable indicator of model performance in a supervised setting when evaluating in a new domain. With respect to input distribution shift, while the word frequency distribution's correlation with model performance is the lowest, it is still significant ($P = .042$). The MMD results reveal a moderately strong correlation ($P = 0.002$). This indicates that in an unsupervised setting, MMD testing with sentence-level representation distribution can be used to estimate model performance when transferring between domains.

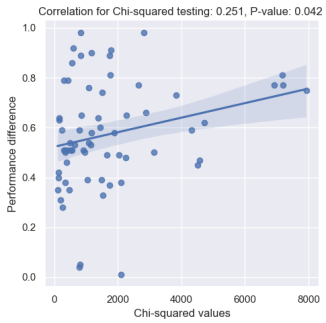

Figure 2.7: Plots for Chi-squared measures with word frequency input distribution and performance difference. Linear regression model fitted.

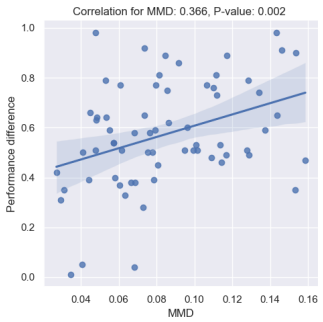

Figure 2.8: Plots for MMD measures with sentence-level input distributions and performance difference. Linear regression model fitted.

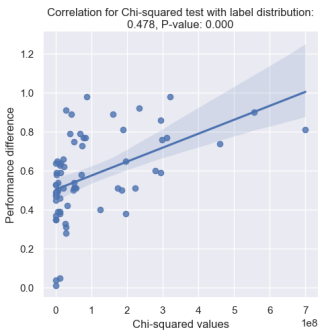

Figure 2.9: Plots for Chi-squared measures with label distribution and performance difference. Linear regression model fitted.

2.5 CONCLUSION

In this chapter, we investigated input data and label distribution shifts across 12 benchmark NER datasets. We compared two different types of representations for input shifts. We systematically measured the shifts using the lens of statistical testing. We measured performance differences by fine-tuning BERT models and calculating the correlation between shifts and performance.

The results show that both word frequency distribution and sentence-level distributional representations are useful for ascertaining shift. Changing between domains results in measurable differences in distribution shifts. Results show that label shift correlates more significantly with performance degradation than input shifts for NER. However, there is still a correlation between input shift and performance degradation. Here, sentence-level representations provide more signals for the relation between distribution shift and performance.

Based on these results, we believe that shift detection and the measurement of distribution shifts can play important roles in tackling NLP tasks, especially for new and low-resource domains. In particular, when applying a model to a new domain or as data changes, the measures detailed above can help researchers and practitioners decide whether the expense of gathering new annotated data and subsequent fine-tuning is warranted. In the future, we hope that distribution shift measurement can become part of widely used NLP paradigms such as crowd-sourcing and active learning.

Now that we see how BERT-based models are not sufficient in the presence of distribution shift, we then continue investigating the second dimension of the challenge, temporal drift.

3

EVALUATION OF UNSUPERVISED STATIC TOPIC MODELS EMERGENCE DETECTION ABILITY

In this chapter, we turn our attention to the second dimension of the challenges identified in Part I: temporal drift. Specifically, we investigate the difficulty of detecting emerging topics in dynamic corpora. Topic models are widely used for analyzing trends and extracting information from conversations in an unsupervised manner. However, while existing evaluation metrics typically assess topic coherence or perplexity, they often overlook the model's ability to detect emerging topics over time. This leads us to the following research question:

RQ2 *To what extent can pre-trained topic models perform topic emergence detection?*

To address this question, we propose a novel, unsupervised evaluation metric that quantifies a model's ability to detect topic emergence without requiring labeled data. We compare representative models from three major categories of topic modeling approaches: Co-Words Clustering (algebraic), Latent Dirichlet Allocation (probabilistic), and BERTopic (neural), across three datasets, ACL Anthology, Web of Science (WOS), and Enron emails.

3.1 INTRODUCTION

Topic extraction, also known as topic modeling, is the process of automatically identifying and extracting meaningful latent concepts or topics from a collection of documents or texts [11]. It is a fundamental technique in natural language processing. Unsupervised topic extraction is commonly used because it does not rely on predefined categories or labeled data.

Topic emergence refers to the appearance of new groups of words representing a topic within textual data. New topics frequently emerge as fields and communities change and it is useful to be able to detect these new topics. For example, science progresses on the principle that new and emerging topics overshadow the older ones [105]. In a social context, organizations are keen to be updated with the new trends that could affect their operations, internally through email [211] and externally using social media platforms [32]. Therefore, understanding which topic model is capable of better detecting topic emergence is beneficial for many applications in academia, entrepreneurship, and government.

Recent advances in topic modeling have focused on developing models that extract topics that are more coherent, diverse, and interpretable [1, 141]. However, there is limited research comparing the ability of these models to detect emerging topics. In other words, when extracting topics and analyzing their

evolution over time, we aim to select a model that can identify emerging topics earlier than others. Specifically, we focus on an unsupervised setting where the ground truth of the topics is unknown.

While previous studies have explored forecasting topic emergence during the "embryo" stage using revolutionary networks and citation-based models [182], others have analyzed retrospective trend detection by identifying shifts in topic proportions or leveraging change-point detection techniques [17, 168]. However, these models they primarily introduce specific models for topic emergence detection without systematically comparing how well different unsupervised static topic models, such as LDA, Bertopic, and CoWords, detect topic emergence. To address this gap, we evaluate the ability of multiple topic models to detect emerging topics in a unsupervised setting systematically. We propose a generic framework that can be applied to any topic models, and an independent evaluation metric to assess their effectiveness.

Comparing topic models presents several challenges. Ideally, if different topic models were to extract the same topics, we could compare the trends produced by each model and determine which model captures the emergence of a topic earlier by having human evaluators evaluate the associated trends. While this approach can be effective, there are still limitations:

1. Different topic models possess different assumptions and initialization procedures, which may lead to different levels of granularity in the learned topics. As a result, there might not be a one-to-one correspondence between topics across different models.

2. The lack of ground truth makes it difficult to apply a systematic comparison approach.

To address these limitations, we propose two evaluation methods. The first involves human evaluation of top-matched topics across models, which provides a qualitative analysis of the generated topics and when they emerge. The second is a quantitative, independent measure of a model's ability to detect emerging topics without ground truth. This measure leverages the generative nature of each model and uses the predictions of emerging topics from a global model as a silver standard for comparison. The measure uses the global model as a benchmark that captures the overall representations of topics, allowing us to assess whether local models accurately reflect the emergence of topics when trained on a subset of data.

To this end, our pipeline consists of four main steps: topic extraction, topic matching, qualitative analysis, and independent emergence performance measure. In the topic extraction step, we input a set of timestamped documents and a static topic model, producing a list of topics with their evolution over time and a trained model as outputs. For topic matching, we use two methods to indicate the prevalence between any pair of topics: Top Words Overlap Rate, based on extracted top words, and the Kullback-Leibler(KL) divergence, based

on topic-document probability distribution. In the qualitative analysis, we inspect the extracted top words. Lastly, we develop an emergence performance measure that can be applied independently across static topic models. This measure uses a global model trained on the entire dataset as a silver standard and compares its predictions with those from local models trained on time-based snapshots of the data. While each generative model can produce topics and assess whether they are emerging, aligning topics from different models is challenging. To overcome this bottleneck, we use documents as a proxy and calculate the agreement between the global and local models on whether a document is emerging.

In this work, we analyze three classic static topic models, which are widely adopted in real-world applications: co-word clustering [21, 22], Latent Dirichlet Allocation (LDA) [13] and Bertopic [72]. They also represent three prominent categories of topic models: algebraic, probabilistic, and neural [1]. Co-word clustering utilizes word co-occurrence patterns, LDA leverages probabilistic modeling, and Bertopic employs neural embeddings to capture rich topic representations. We compare extracted topics across different models. We also exhibit and compare different matching strategies across different models and present topic emergence among different methods.

To perform these comparisons, we conduct comprehensive experiments on three datasets: Web of Science bio-medical publications, ACL anthology publications [177], and the Enron email dataset [102]. These datasets cover diverse domains and contain varying degrees of topics. Examining models on these datasets allows us to compare their effectiveness and generalizability across contexts with varying levels of topic institutionalization.

The development of measuring and comparing the topic model's ability to topic emergence in the NLP community can be of help to different communities, including firms (company-wide emails), governments and scientists (scientific abstracts) [107], in choosing the model for detecting changing topics. Additionally, this framework can be applied to patent analysis and other types of text analysis [139]. Another application area is innovation policy, where the identification of emerging/changing scientific topics is of great interest to governmental policymakers, which can be used to foster national-level scientific competitiveness [184].

Summarizing, the main contributions of our work are as follows:

- *A systematic framework for topic emergence detection.* We introduce a comprehensive pipeline that extracts topic, match topics across different models and track them for their emergence. Unlike previous studies that focus on a single topic modeling, our framework enables direct comparison of multiple models in an unsupervised setting.
- *The first comparative study for static topic models for detecting emerging topics.* We perform a qualitative analysis comparing how three widely used topic models capture emerging topics. This is the first study that systematically

benchmarks different static topic modeling approaches for emergence detection.

- *A novel evaluation metric for topic emergence detection.* We propose an unsupervised metric that assesses a model's ability to detect emerging topics without the need for manual assessment. This metric enables scalable evaluation across diverse textual domains.

3.2 RELATED WORK

With respect to topic extraction, prior work has typically relied on co-word clustering [6, 35, 234] and probabilistic topic models like LDA [12, 27, 125, 163] to extract topics from textual data (e.g., scientific publications, emails) and track their evolution through time. Recent advances, particularly the development of pre-trained large language models like BERT [54, 56, 72], have paved the way for new and promising applications in topic extraction.

Evaluating topic models presents a multifaceted challenge that hinges on both the accuracy and the relevance of the topics generated. Traditional metrics such as perplexity and coherence scores have been foundational, providing quantitative benchmarks that assess the internal consistency and semantic similarity within topics [152, 176]. However, these metrics often fall short in capturing the practical utility of the topics in real-world applications. More recent approaches emphasize user-centric evaluations, where the interpretability and applicability of topics to specific tasks are assessed through user studies or expert validations [26, 108]. Such methodologies aim to bridge the gap between statistical performance and practical significance, ensuring that the topics are not only coherent but also meaningful and actionable in specific contexts. Despite these advances, there is very limited development in evaluating topic model's ability to detect topic emergence.

Within the literature on topic modeling, there is a line of work that focuses on Dynamic Topic Modeling (DTM) [12], which explicitly incorporates the time associated with the documents in the modeling process. The topic representation at timestamp t depends on the representation at $t-1$. DTM focuses on how topic representations evolve over time, offering different representations given time slices. Although DTM is capable of modeling how the concept of a topic shifts, it does not assist in detecting topic emergence. The downstream applications of such models often focus on tracking the dynamics or development of topics [69, 113] instead of detecting emergence. In contrast to our work, we use a retrospective approach focused on emergence that can learn a global representation of the topic and compare it with local representations, using a document-level proxy for emergence detection.

Recent work on emerging topic detection has explored both forecasting-based and retrospective approaches. Forecasting-based methods aim to predict the rise of emerging topics before they become widely recognized, leveraging his-

torical trends and external signals. AUGUR [182] introduced a forecasting-based approach leveraging evolutionary networks to predict topic emergence before their recognition. Other forecasting methods utilize incremental topic modeling [65] and anomaly detection in topic distributions [174] to detect early signals of emerging research trends. These approaches typically rely on time-series modeling, citation networks, or machine learning-based trend extrapolation to anticipate future developments.

On the other hand, retrospective approaches focus on identifying when topics have already emerged in historical data. Methods such as BERTrend [17] and ATEM [168] apply neural topic modeling and graph-based embeddings to detect topic shifts in past corpora. Additional studies explore word embedding trajectory monitoring [39] and structural changepoint analysis [15, 215] to identify when new topics emerge based on shifts in semantic space or statistical deviations in topic prevalence.

However, these methods typically introduce a single model without benchmarking multiple topic modeling techniques, making it difficult to compare the relative effectiveness of different approaches. Furthermore, other works on topic model evaluation primarily focus on coherence and perplexity [99] or even topic coverage [104], yet these metrics fail to capture a model's ability to detect emerging topics. In contrast, our work systematically compares three widely used static topic models (LDA, BERTopic, and CoWords) for emergence detection across multiple domains and introduces an independent evaluation metric that does not rely on external metadata or predefined ground truth labels.

Similar to our work, novelty detection also uses documents as a proxy for change. Novelty detection aims to find text with new information compared to what has been seen or known before [67]. Prior studies frame novelty detection as a document-level binary classification problem, where documents are classified as "novel" or "not novel" based on a set of existing documents [66, 150, 180]. Our work is different in that we first identify emerging topics using a trained model subsequently, we determine if a document is emerging if it is associated with an emerging topic. This approach differs from assessing if topics are evolving based on a single model [197].

3.3 TOPIC EXTRACTION PIPELINE

To study method performance on emergence detection, we develop a pipeline that takes a dataset with time as input and then outputs matching topics and their change over time. Our general framework is shown in Figure 3.1. The pipeline consists of these steps: data preprocessing, topic modeling, topic and trends extraction, topic matching, and finally, matched topics emergence analysis.

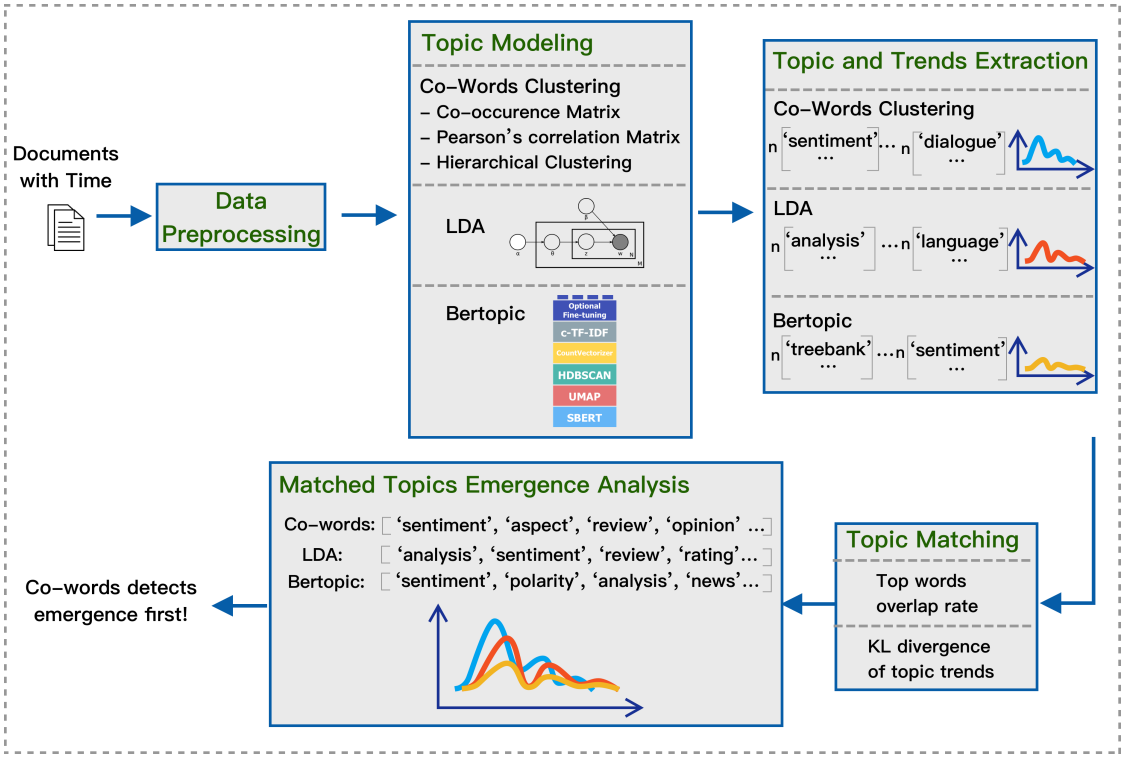

Figure 3.1: Pipeline for extracting and matching topics with three models.

The goal of the pipeline is to generate and compare topics with their trends using varying static topic models in an unsupervised manner. The final output will be topics with their associated trends that are generated and matched using different topic models. Once the matched topics and their trends are generated, we can then apply qualitative analysis and quantitative analysis on which model is better at emergence detection.

3.3.1 *Overview of the pipeline*

The input of the pipeline is a set of documents with a time stamp for each document. First, we apply the preprocessing procedure for each dataset following Natural Language Processing practices. We then train each model on the processed texts to generate topics. For each topic model, we fix the number of topics for each dataset to 100. Any semantic space can be divided into an unlimited number of latent spaces. A topic can also be divided into different latent concepts. We fix the number of topics to attempt to ensure the statistical and semantic meaningfulness of the topics and a balanced trade-off between topic variation and interpretability. We use the number 100 because it is aligned with previous study [197].

After the models are trained with 100 topics, we subsequently extract 20 top words for each topic as the representation of that topic. The top words

are selected based on frequency to ensure they are diverse and representative. We then apply a hierarchical topic-matching strategy based on two metrics: top words overlap rate and KL divergence over the matched trends. After the topics are matched, we plot the trends and then apply both qualitative evaluation and quantitative evaluation on topic emergence detection ability. The code and data used in this paper are available at: `https://zenodo.org/records/14503316`. We now explain each of the steps in details in the following sections.

3.3.2 *Data Preprocessing*

To reduce the noise coming from uninformative words, the raw text of the merged titles and abstracts has been pre-processed. Pre-processing can drastically impact extraction quality. Hence, we describe our entire preprocessing process in the following seven steps with detailed parameter specifications.

Step 1. The date/year of the document, the title of the document and abstracts/email bodies of the document are needed for further analysis. Therefore, we remove entries where any of these fields are empty. Additionally, for computational reasons, we remove documents where the length is more than 700 tokens for the Enron dataset.

Step 2. Each document is tokenized and converted into a list of lowercase tokens. Tokens shorter than two characters and longer than 15 are discarded in the tokenization process. Accents and punctuation are removed.

Step 3. Documents are further de-noised by removing English stopwords, i.e., frequently used words that do not provide significant distributional information.

Step 4. The vocabulary is enriched with bigrams, i.e. pairs of consecutive tokens that often appear together (e.g. “amino acid”, “frontal cortex”, etc.). We use Point-wise Mutual Information (PMI) as a score function and a threshold of 100, and the minimum collective frequency for a valid bigram is 20.

Step 5. We perform lemmatization to group the inflected forms of a word in a single token.

Step 6. After performing part-of-speech tagging, we keep only nouns, adjectives and verbs.

Step 7. We removed from the remaining tokens all the words that occur in more than 25% of the documents or in less than 0.01% of the documents.

3.3.3 *Topic Modeling*

We select three models for comparison: 1) co-word clustering, 2) LDA, and 3) Bertopic. We follow the categorizations from [1], selecting the most widely used topic model from the three prominent categories: algebraic, probabilistic, and neural. Comparing these three models provides us a intuition on how the three types of topic models perform on topic emergence detection.

CO-WORD CLUSTERING. The idea behind co-word clustering is that the co-occurrence of words describes the contents of the texts [21]. Based on this notion, methods have clustered words in the keywords lists, titles and abstracts, or other publication data fields, using multivariate statistical techniques, such as factor analysis, principal component analysis, and hierarchical clustering to obtain topics [217]. In this work, we utilize hierarchical clustering for topic modeling.

LDA. As a probabilistic topic model, the basic assumption of LDA is that the words are generated according to a mixture model where the mixture proportions are random, and the mixture components or topics are shared by all documents [12]. It is based on the idea that documents contain multiple topics, intended as distributions over a fixed vocabulary [11, p. 78]. However, one limitation of LDA is that it models natural language as bag-of-words, discarding the word orders in the document.

BERTOPIC. As one of the most widely used neural topic models, Bertopic is popular for its adaptability. It utilizes the recent development in NLP, modeling sentences with word orders using sentence embedding models such as sentence-BERT.

The semantic space created by sentence-BERT can be seen as a continuous space of sub-topics. We can discretize this space by detecting high-density areas and associating them with a topic. To do so, we use dimensionality reduction such as UMAP and clustering techniques such as HDBSCAN. UMAP is a manifold learning technique that is good at preserving both global structure and local structures.HDBSCAN [24] is a density-based, hierarchical clustering method that is noise-aware (i.e., potentially outliers aren't forced to belong to a cluster and could be labeled as noise) and based on soft assignment (i.e., each point is associated to its cluster with a confidence score). Moreover, HDBSCAN relaxes the need to set the number of clusters as a hyperparameter, requiring specifying only the minimum number of clusters desired. Once the clusters have been identified, we can retrieve topic embeddings by calculating the mean of all the documents belonging to the same cluster, i.e., the centroids. Since we learn word embeddings and document embeddings jointly in the same semantic space, we can look at the K words closer to the centroid to get a set of representative terms for that topic. This process is called fine-tuning the topic representation. Specifically, in our work, we use the Maximal Marginal Relevance algorithm to reduce the redundancy of the extracted keywords in each topic.

After extracting topics with each method, we observed that some topics have few documents associated with and some topics have only digits as top words. We filter these outliers out.

3.3.4 *Topic and Trends Extraction*

Once the model is trained, topics can be extracted using different representations. In this work, we represent each topic by its top words, providing an interpretable summary of the topic's content. Additionally, we track the evolution of each topic over time by quantifying the number of documents associated with it at different time points. This allows us to analyze topic trends, identifying patterns of emergence and decline across the dataset. However, the absolute number of documents associated with a topic may not be a reliable indicator of its popularity, as the total number of documents per year varies. To address this, we use prevalence as a normalized measure of topic popularity. Prevalence quantifies topic prominence by dividing the number of documents assigned to a topic by the total number of documents in that year, ensuring a fair comparison across different time periods. This normalization allows us to accurately compare topic trends over time, independent of fluctuations in document volume.

3.3.5 *Topic Matching*

Once the topics and their associated trends are extracted, comparing topics across different models requires a robust matching strategy. Since topic models may learn different representations of the same underlying concepts, we need to align topics from different models to conduct a fair evaluation of their ability to detect emerging topics.

We employ two matching strategies in this work:

- **Top Words Overlap Rate (TWOR)**: for matching topics based on their most representative words.
- **Kullback–Leibler (KL) divergence**: for matching topics based on their temporal trends.

The intuition behind TWOR is that topics are primarily characterized by their most representative words, making an exact match based on word overlap a straightforward way to determine topic similarity. Meanwhile, KL divergence quantifies the similarity of topic prevalence distributions over time, under the assumption that similar topics exhibit similar temporal trends.

TOP WORDS OVERLAP RATE The first matching strategy is based on comparing the top words of each topic. For each extracted topic, we retrieve the top n words that are most representative of that topic. These words are selected based on Term Frequency-Inverse Document Frequency (TF-IDF) scores, which measure a word's relevance in the corpus.

In Bertopic, a cluster-level TF-IDF variant (c-TF-IDF) is used, calculating the weights as term importance at a cluster level:

$$\text{ctfidf}(t, c) = \text{tf}(t, c) \cdot \log\left(1 + \frac{A}{f_c}\right), \tag{3.1}$$

in which t is the term, c is the cluster, $\text{tf}(t, c)$ is the term frequency of term t within cluster c, indicating how often term t appears within that cluster. A is the average number of words per cluster.

After obtaining the n top words for each topic T, we calculate the overlap rate between any two topics:

$$\text{TWOR}(i, j) = \frac{T_i \cap T_j}{N}, \tag{3.2}$$

where N is the number of words for each topic.

KL DIVERGENCE The second score function is Kullback–Leibler divergence for topic trends. This function is selected under the assumption that similar topics will have similar trends over time. We calculate topic prevalence based on the number of documents associated with the topic given any timestamp:

$$\text{Prevalence}(t_i, y_j) = \frac{N_{ij}}{N_j}, \tag{3.3}$$

in which N is the number of documents given topic i and year j. We then apply KL divergence between two distributions:

$$D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{3.4}$$

HIERARCHICAL TOPIC MATCHING STRATEGY Because topic models differ in how they define topics, there is no direct one-to-one correspondence between topics across models. Instead, each topic from one model can have partial overlap with multiple topics from another model, creating an N-to-N mapping problem. To address this, we use a hierarchical topic matching approach:

- First we match Bertopic topics (T_{bertopic}) to LDA topics (T_{lda}) using one of the above matching methods (TWOR or KL divergence).
- Next, we match CoWords topics ($T_c w$) to the already aligned Bertopic-LDA topics ($T_{\text{lda\_bertopic}}$), instead of directly matching all three models at once.
- To resolve conflicts in the N-to-N mapping, we prioritize matches by selecting topics that maximize the sum of matching scores, ensuring that the most representative topics are aligned across models:

$$\max \sum (S_{\text{lda\_cw}}, S_{\text{lda\_bertopic}}). \tag{3.5}$$

Through this hierarchical strategy, we reduce complexity and maintain coherence in topic alignment, making it possible to systematically compare the models' ability to detect emerging topics.

3.3.6 *Matched Topics Emergence Analysis*

We employ two analysis for evaluating models' emergence detection ability: qualitative approach and quantitative approach.

3.3.6.1 *Qualitative Comparison*

Although all these methods have achieved outstanding performance on a variety of datasets, comparing them directly on the matched topics is still challenging. When the one-to-one mapping of topics exists across models, then we can directly compare their evolutions and derive the ability for early emergence detection. However, different topic models exhibit different assumptions and initializations, often resulting in the extraction of topics with different levels of granularity. For example, in the NLP context, one topic extraction model might extract topics with top words such as *['dependency', 'output', 'tree', ...]* and *['parse', 'syntactic', 'class', 'contextual', 'syntax',...]* which corresponds to two different NLP tasks "dependency parsing" and "syntactic parsing", a different model might extract only one topic with top words *['structure', 'parse', 'syntactic', 'dependency', ...]*, which represents the parsing task in general. Therefore, we perform a topic matching with two distinct score functions, Top Words Overlap Rate and KL-divergence, and then evaluate the top-matched topics qualitatively.

3.3.6.2 *Quantitative Evaluation Metric*

In this section, we describe our proposed approach for measuring a model's ability to detect emerging topics without relying on ground truth labels. Unlike traditional topic modeling evaluations that focus on coherence scores or perplexity, our approach quantifies how well a model captures topic emergence over time by assessing the agreement between local and global models. This method is summarized in Figure Figure 3.2.

Detecting emerging topics is challenging due to the lack of predefined ground truth labels that indicate when a topic becomes significant. Existing evaluation metrics such as topic coherence measure the semantic quality of topics based on word co-occurrence patterns, while perplexity evaluates how well a model predicts unseen text. However, these metrics do not assess whether a model effectively identifies when a topic is emerging. Our approach addresses this limitation by introducing a framework that compares local and global topic models to infer a model's emergence detection ability.

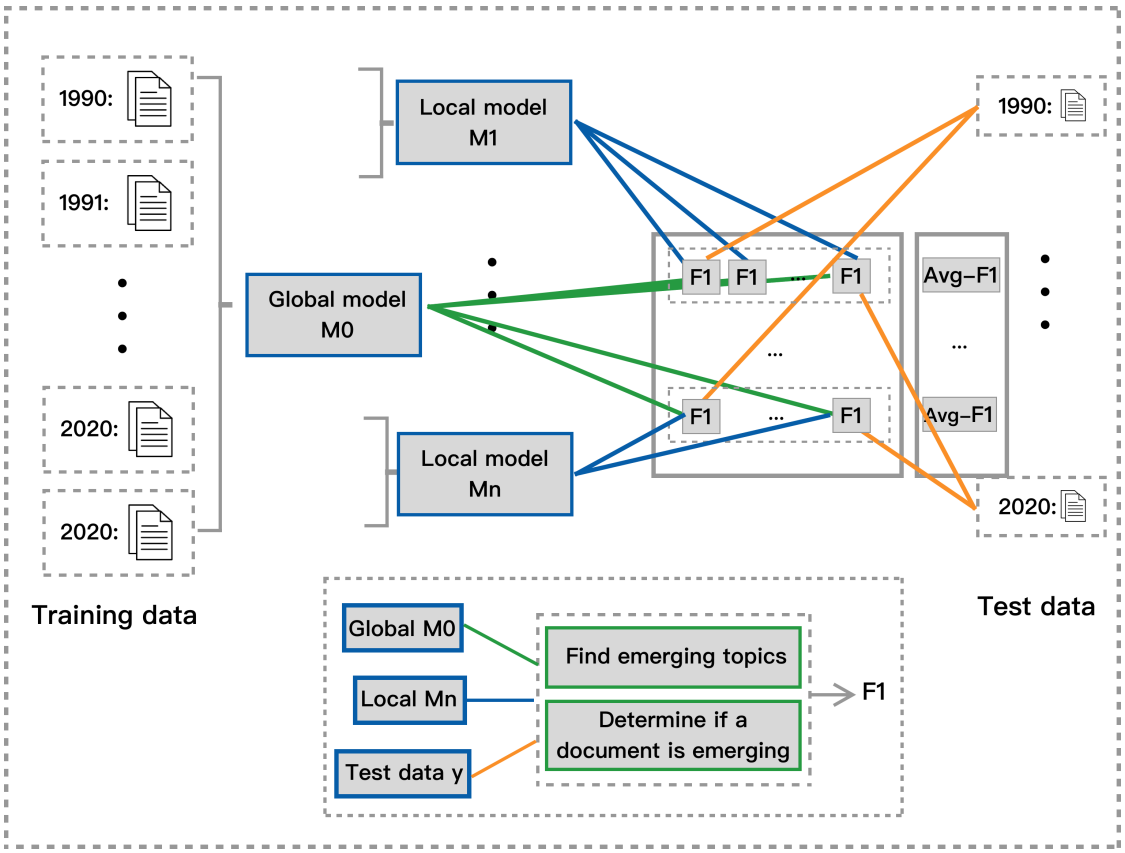

Figure 3.2: Independent quantitative measure between global and local model.

To evaluate a model's ability to detect emerging topics, we train a global model on the full dataset and local models on segmented time slices and then compare their predictions. The global model serves as a silver standard, enabling us to assess how well each local model captures emerging topics within its respective time period. As illustrated in Figure Figure 3.2, the orange lines represent test data from a given time span, the green lines indicate emerging documents classified by the global model, and the blue lines indicate emerging documents classified by the local models.

We begin by segmenting the dataset based on timestamps and splitting each segment into training (80%) and test (20%) data. We then train a global topic model on the entire dataset to provide a comprehensive reference for topic emergence. Simultaneously, we train local topic models on each time segment's training data and evaluate them on test data from all other segments. We compute precision, recall, and F1 scores to measure the agreement between local and global models, providing a quantitative assessment of how well each model detects topic emergence.

Since there is no predefined ground truth for emerging topics, we introduce the concept of emerging documents. Rather than directly tracking topic-level trends, we classify a document as emerging when its dominant topics are iden-

tified as emerging by the global model. This enables a systematic evaluation of how well local models align with the global model in identifying emerging topics at a document level.

A higher F1 score between local and global models indicates greater agreement, meaning that the local model successfully captures emerging topics in a manner similar to the global model. Higher F1 scores also suggest that the model is more sensitive to topic changes and better at tracking emerging topics within shorter time spans. In contrast, lower F1 scores indicate that the local model struggles to capture emerging trends visible in the full dataset, implying that it is less responsive to short-term topic evolution. Models with lower F1 scores may be better suited for capturing long-term topic distributions rather than detecting short-term emergence.

Using document-level agreement instead of topic-level comparisons allows us to capture real-world knowledge shifts, making it a meaningful proxy for topic emergence. Since topic models assign probability distributions to documents, tracking the emergence of topics through documents provides a granular assessment of how topics gain prominence over time. Additionally, this method avoids the limitations of direct topic-level comparisons, where different models may produce topics at varying levels of granularity.

While this approach provides a robust framework for retrospective emergence detection, it has some limitations. The effectiveness of the metric depends on the granularity of learned topics—if topics are too broad, emerging documents may be harder to identify. Additionally, the global model's predictions may introduce bias, as it could overfit dominant research trends and fail to recognize smaller, emerging topics. Another limitation is the lack of external validation; while we use the global model as a silver standard, future work could incorporate expert annotations or alternative reference points, such as external event timelines.

Overall, our quantitative evaluation metric provides an objective and scalable method for assessing topic emergence detection. By leveraging document-level agreement between local and global models, we introduce a ground-truth free approach that is applicable across different datasets and topic models.

FINDING EMERGING TOPICS Given a topic trend, determining if a topic is emerging in a certain time period fundamentally relies on if the trend shows an upward trajectory. By definition, a topic emerges when it gains prominence over time, in which the most direct way to quantify it is by computing the growth rate. This aligns with other retrospective methods of identifying emerging topics [65]. While other forecasting-based methods utilizes external information such as citation graphs, their emerging detection methods are not directly applicable in our setting.

In this work, we calculate the growth rate between any given timestamp t_i and t_j; if the growth rate is positive, the topic is emerging; otherwise, it is not

Periods	WoS	Periods	ACL	Periods	Enron
1990-1994	6391	1980-1984	551	1998-01 - 1998-11	68
1995-1999	12094	1985-1989	919	1998-12 - 1999-04	461
2000-2004	19020	1990-1994	1903	1999-05 - 1999-09	4259
2005-2009	29347	1995-1999	2458	1999-10 - 2000-02	18401
2010-2014	40910	2000-2004	3947	2000-03 - 2000-07	50797
2015-2019	50791	2005-2009	7410	2000-08 - 2000-12	115777
2020-2020	12946	2010-2014	11977	2001-01 - 2001-05	136026
		2015-2019	18822	2001-06 - 2001-10	77927
		2020-2022	15212	2001-11 - 2001-12	36103
Total	171499	Total	63199	Total	439819

Table 3.1: Number of documents per time period after pre-processing for three datasets.

emerging. For topic k, we determine if k is emerging in the time period of i and j, we use:

$$g_k(t_i, t_j) = \frac{C_{ik} - C_{jk}}{C_{ik}}, \tag{3.6}$$

in which C_{ik} is the number of documents associated with topic k at time t_i, reflecting how frequently the topic appears in that period. Topic k is emerging is $g_k(t_i, t_j) > 0$, k is not emerging otherwise.

Specifically, for each given time period, we first find the emerging topics. Then, we associate each document with topics based on the predicted probability of the model. If a document is associated with topics that are determined to be emerging by the model, it is an emerging document. We calculate the F1 score based on the agreements between the global and local models on the same set of documents.

3.4 EXPERIMENTAL SETUP

We detail our experimental settings here. We apply our pipeline to three datasets from different domains, using three topic models from different categories. We then match topics and evolutions extracted from three models, and subsequently analysis their topic emergence detection.

3.4.1 *Datasets*

We used three datasets to validate our experiments: Biomedical publications on the Web of Science (WoS), anthology publications of ACL (ACL), and the Enron email dataset. These three datasets cover the biomedical, natural language processing, and corporate communication domains. Both WoS and ACL are scientific publications, presenting more structured language usages, while Enron dataset represent more fuzzy language usage. Additionally, WoS and ACL datasets represent different domains, showing how domains could potentially affect topic extraction. For each dataset, we preprocess the raw text following the steps in Section 3.3.2, and the statistics of each dataset after pre-processing are shown in Table 3.1.

WEB OF SCIENCE The WoS dataset contains 171,499 publications from the *Life Sciences and Biomedicine* field between 1990 and 2020. Wos has been widely adapted to management studies [117, 126]. In our work, we crawl the web and create the dataset till 2020, making sure the corpus is relatively up-to-date. We use abstracts together with the title as our corpus.

ACL The ACL anthology dataset [177] contains publications in the domain of NLP and Computational Linguistics. The publications are conference papers, journal articles and workshop papers spanning from 1980 to 2022. The dataset contains 80,013 documents originally and 63,199 documents after preprocessing. Similarly, we use abstracts and the title of each document.

ENRON The Enron email dataset contains email conversations within the energy company Enron Corporation from 1998 to 2001. The content of the dataset covers topics from business practices to organizational communication. Unlike scientific publications, we use email bodies and email subjects as our corpus. Note that some documents might contain more than thousands of tokens. As mentioned in Section 3.3.2, we remove documents that are longer than 700 tokens for computational reasons, especially for Bertopic.

The two academic publication datasets, WoS and ACL tend to be more formal and have greater structure due to the peer-reviewed publication processes. However, the Enron email dataset contains more casual-style texts, including misspellings, informal language usage, etc. Additionally, the length of abstracts is often between 150-250 words, while the length of documents in emails can vary drastically.

We adopt different pre-trained sentence-bert models due to varying domains. We apply biomedical sentence-bert for the WoS dataset and the distilled-sentence-roberta model for the ACL and Enron datasets.

To sum up, in our experiments, we apply CoWords, LDA, and Bertopic to each dataset and extract topics over time. We then perform topic matching

between models using top-word overlap (TWOR) and KL divergence to align equivalent topics across different methods. Once topics are matched, we analyze their prevalence trends to assess how each model detects topic emergence. Finally, we evaluate the models using both qualitative trend analysis and a quantitative emergence detection metric, which measures the agreement between local models trained on time-segmented data and a global model trained on the full dataset. This setup allows us to systematically compare the ability of different topic models to detect emerging topics across domains. We present the results of these experiments and discuss them in the following sections.

3.5 RESULTS

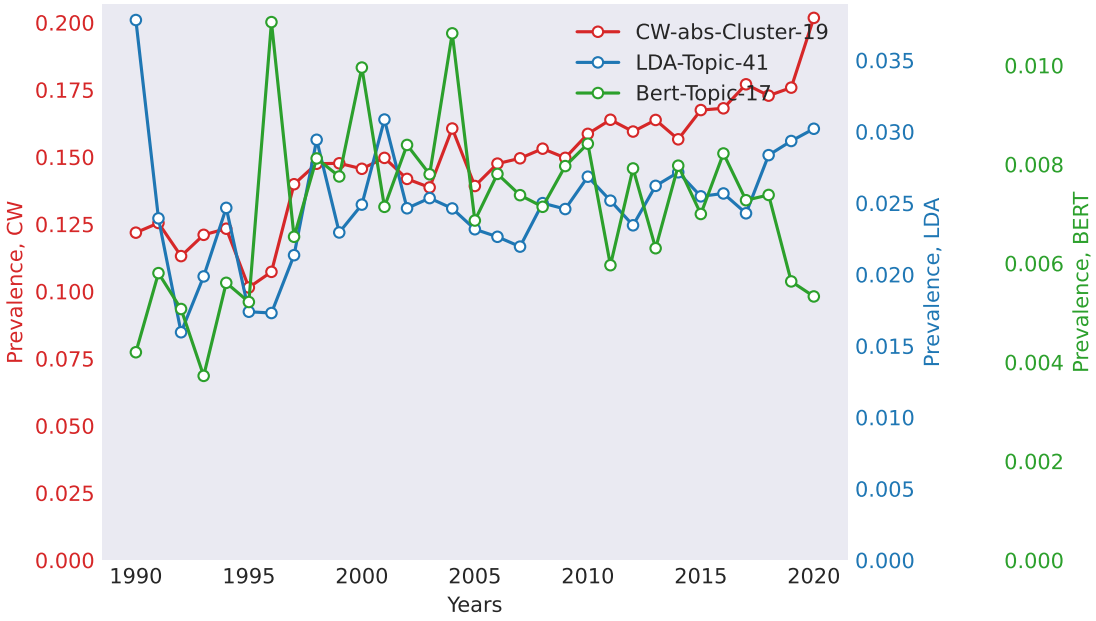

Figure 3.3: Selected match for WoS. The top 10 words for each method are as follows. CoWords: {infection, mortality, transmission, virus, spread, infect, vector, incidence, viral, epidemic}, LDA: {mouse, infection, antioxidant, virus, observed, stimulation, respond, protection, viral, infect}, Bertopic: {epidemic, infection, virus, viral, vaccination, model, transmission, vaccine, infectious, infect}

This section presents the results of applying all three topic models (CoWords, LDA, and Bertopic) to each dataset. We first compare the TWOR matching and KL divergence qualitatively by examining the top-matched topics manually. We then present a quantitative evaluation using our proposed metric to measure the models' ability to detect emerging topics.

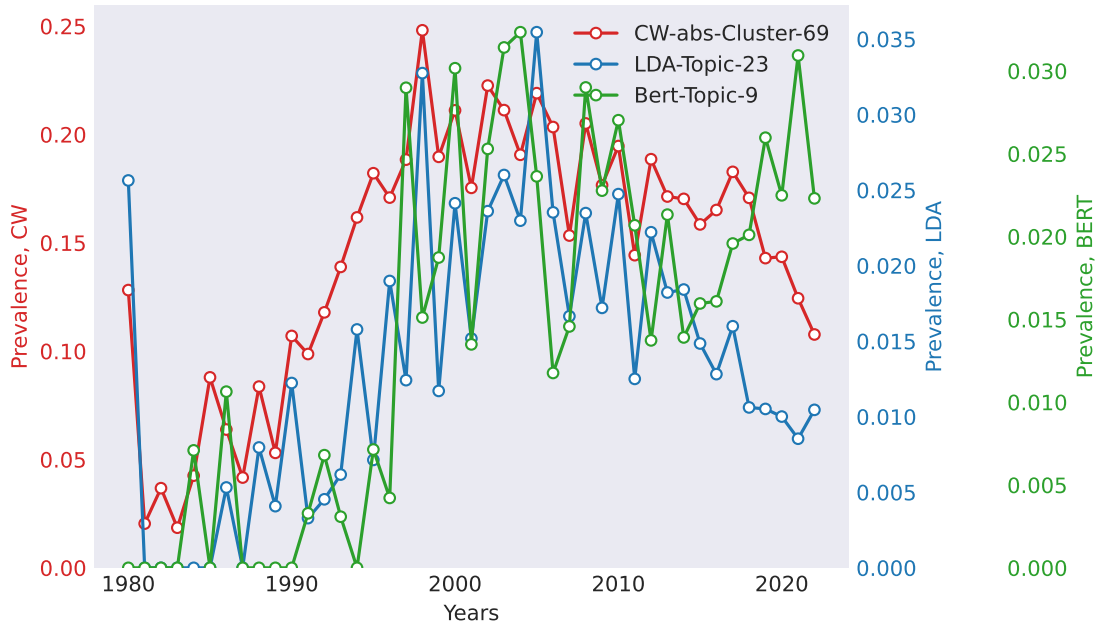

Figure 3.4: Selected match for ACL. The top 10 words for each method are as follows. CoWords: {segmentation, tag, chinese, character, segment, boundary, tagging, tagger, wordlevel, partofspeech_tagged}, LDA: {accuracy, segmentation, character, segment, rich, morphology, morpheme, unsupervise, convention, lefttoright}, Bertopic: {segmentation, chinese, word, character, model, tagging, partofspeech_tagged, ngram, language, accuracy}

3.5.1 *Qualitative Analysis*

To compare TWOR and KL divergence, we examine their top-matched topics across models and datasets. Table 3.5 presents the topics with the highest matching scores for using each topic model, applying two matching strategies.

Our analysis shows that TWOR consistently retrieves semantically coherent topics, whereas KL divergence often prioritizes temporal trends over direct topic alignment. For example, in the WoS dataset, TWOR matches a topic centered on Diabetes (e.g., "insulin," "glucose," "obesity"), while KL divergence instead retrieves bioelectromagnetic topics, likely due to similar growth patterns rather than shared semantics. A similar pattern emerges in the ACL dataset, where TWOR retrieves Grammatical Error Correction, while KL divergence retrieves broader methodological terms (e.g., "baseline," "metric"). These results highlight that TWOR is better suited for direct topic alignment, while KL divergence may be more useful for finding temporally related but semantically distinct topics.

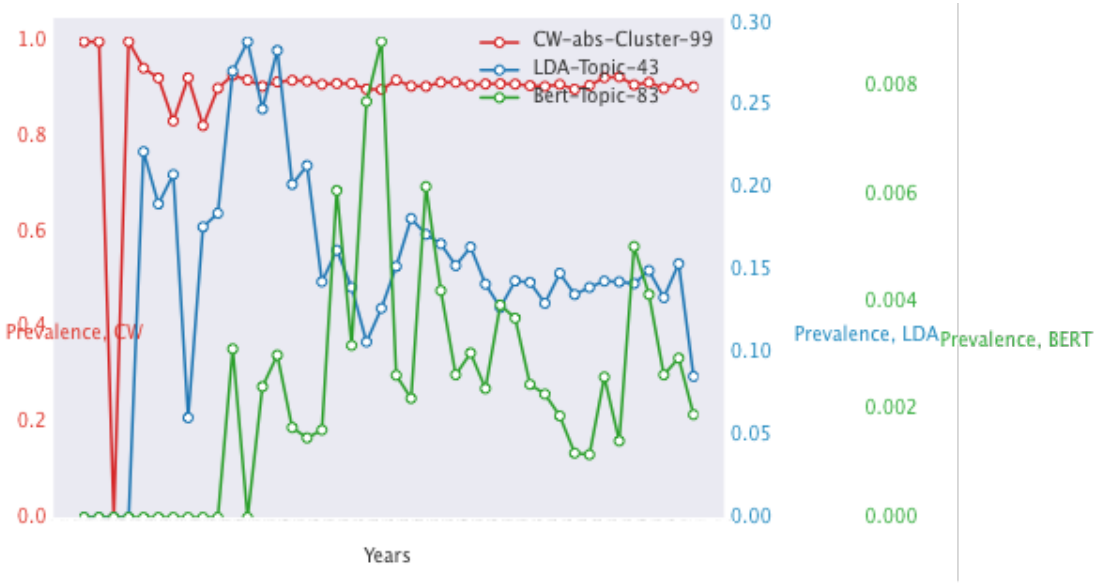

Figure 3.5: Selected match for Enron. The top 10 words for each method are as follows. CoWords:{email, agreement, question, receive, meeting, schedule, request, contact, file, list}, LDA: {agreement, contract, review, document, draft, bind, title, attorney, signature, wrong}, Bertopic: {abb_transformer, abb, existence, agreement, override, transformer, signature, initial, word, option}

3.5.1.1 *Case Study: Topic Evolution Across Domains*

In this section, we present a case study to analyze how different models track emerging topics over time across the three datasets. We analyze the topics based on their extracted top words and their corresponding revolution.

CASE 1: BIOMEDICAL RESEARCH (WOS) - THE RISE OF EPIDEMIOLOGY The WoS dataset captures the emergence of epidemiology-related topics, as shown in Figure 3.3. TWOR-aligned topics (e.g., "insulin," "glucose," "obesity") reflect a clear trend in diabetes research, with increasing prevalence over time. This aligns with global public health concerns and real-world epidemics like Ebola and SARS, which drove research interest in epidemiology. In contrast, KL divergence retrieves a bioelectromagnetics-related topic, demonstrating its tendency to group topics with similar temporal patterns rather than semantic alignment.

CASE 2: COMPUTATIONAL LINGUISTICS (ACL) - ADVANCES IN WORD SEGMENTATION The ACL dataset provides insight into the development of Natural Language Processing (NLP) over time (Figure 3.4). The matched topic reflects Grammatical Error Correction and Word Segmentation, showing a surge in research interest in the late 1990s and early 2000s, aligning with the rise of statistical NLP approaches. This suggests that topic models effectively track re-

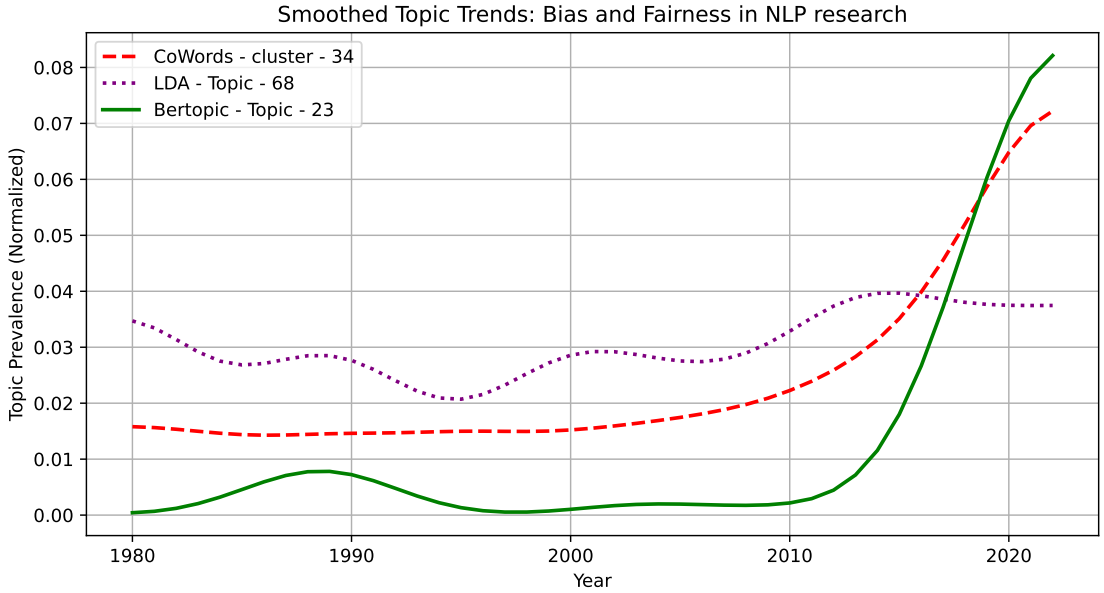

Figure 3.6: Smoothed trends for the topic: Bias and Fairness in NLP research. CoWords:{bias, gender, mitigate, age, demographic, biased, debiase, fairness, female, male}, LDA: {gender, mitigate, transe, people, production, rapidly, expectation, game, progress, produce}, Bertopic: {gender, pronoun, bias, pronoun_resolution, stereotype, female, debiase, language, gendere, stereotypical}

search trends, reinforcing the usefulness of retrospective analysis in scientific trend forecasting.

CASE 3: CORPORATE COMMUNICATION (ENRON) - VARIABILITY IN BUSINESS AGREEMENTS The Enron dataset presents a unique challenge due to its unstructured, informal text (Figure 3.5). The matched topic centers around business agreements, but topic volatility differs across models. LDA and CoWords extract broader business-related terms, whereas BERTopic produces more volatile topic representations, possibly due to its reliance on sentence embeddings, which capture finer-grained variations in contract-related discussions.

CASE 4: A EMERGING TOPIC CASE - BIAS AND FAIRNESS IN NLP RESEARCH We present a case study where our models detect an emerging topic related to Bias and Fairness in NLP research. This topic matched across models, demonstrates how different topic modeling approaches capture the emergence of new research directions over time. Figure 3.6 presents its smoothed prevalence trends. We observe that Bertopic detects weak signals first (before 2000) but does not show sustained growth until post-2010, suggesting early semantic awareness but delayed recognition of topic prevalence. CoWords and LDA capture sustained emergence earlier, with CoWords showing a gradual increase

	WoS - Overlap	**WoS - KL divergence**
CoWords	liver, glucose, lipid, insulin, fat, diabetic, hepatic, obesity, fatty_acid, cholesterol	absorption, mobile, bioelectromagnetic, guideline, electromagnetic, wileyliss, mobile_phone, antenna, phone, mhz
LDA	evaluate, assess, estimate, glucose, consistent, plasma, obesity, estimation, differential, trial	model, spatial, paper, focus, derive, finally, account, mathematical, temporal, input
Bertopic	diabetic, insulin, glucose, rat, diabetes, mouse, insulin_resistance, adipocyte, islet, hepatic	cardiac, muscle, cardiomyocyte, heart, calcium, mouse, myocardial, cell, channel, skeletal_muscle

	ACL - Overlap	**ACL - KL divergence**
CoWords	error, correct, grammatical, correction, spelling, spell, gec, error_correction, spelling_correction, misspell	approach, base, set, compare, technique, accuracy, perform, apply, label, combine
LDA	detection, error, detect, translate, correct, correction, check, sensitive, spelling, loglinear	method, approach, propose, evaluate, score, focus, exist, baseline, outperform, metric
Bertopic	error, grammatical, correction, gec, learner, error_correction, spelling_correction, spelling, detect, chinese	dialogue, speech, dialog, speak, conversation, response, conversational, agent, recognition, speaker

	Enron - Overlap	**Enron - KL divergence**
CoWords	time, deal, gas, market, power, service, price, day, energy, company	ect, subject, message, forward, original, fax, hou, tomorrow, lon, confirmation
LDA	price, offer, risk, lock, sfodenver, rls_tariff, vjw, faithbase, obliterate, farreache	message, original, delete, civil_libertie, scarff, clintonappointe, groyer, faithbase, obliterate, bridget_maronge
Bertopic	ect, email, subject, message, service, market, receive, gas, business, contact	subject, ect, message, forward, email, agreement, attach, meeting, market, issue

Table 3.5: Extracted topics with the highest score using two matching strategies.

from the early 2000s, while LDA detects a structured rise between 1995 and 2005. CoWords excels in tracking early co-occurrence shifts, LDA tends to stabilize and capture steady trends, and Bertopic is more sensitive to semantic shifts once the topic is widely established. The detected trends pattern aligns with real-world topic adoption trends. Early discussions on NLP bias existed before 2000, but they were scattered and lacked formal structure, similar to how Bertopic detects weak signals first. Between 2000 and 2010, fairness in NLP gained academic traction, reflected in LDA's structured rise. Finally, post-2010, NLP bias became a mainstream issue, driven by ethical AI debates and policy discussions, aligning with Bertopic's stronger detection at this stage.

From the qualitative analysis we observe the potential characteristics of each topic model for topic emergence ability. However, they lack systematic comparisons and quantified measure for evaluating their performance for the task. We then perform the quantitative analysis using our proposed metric.

3.5.2 *Quantitative Analysis*

Next, we evaluate the models' ability to detect emerging topics using our proposed evaluation metric. Since CoWords model does not generate document-topic probability distribution, the evaluation metric is not applicable. Figure 3.7 presents the average F1 scores between local models trained on different time segments and a global model trained on the full dataset. Results indicate that LDA achieves a higher average F1 score (80.1%) compared to Bertopic (56.6%), suggesting that LDA more consistently detects emerging topics over time. Furthermore, results also show that F1 scores increase as segment size grows, suggesting that models trained on longer time spans capture more stable topic representations.

To further analyze how local models align with the global model, we visualize F1 score heatmaps in Figure 3.8 and Figure 3.9. We have a few key observations from the heatmaps. First, LDA achieves more stable F1 scores across time compared to Bertopic. Second, Periods of lower F1 scores (e.g., 1980s, post-2015) indicate time spans where emerging topics were harder to detect. Third, Bertopic's lower consistency suggests it is more sensitive to novel topics, while LDA generalizes better even with limited training data.

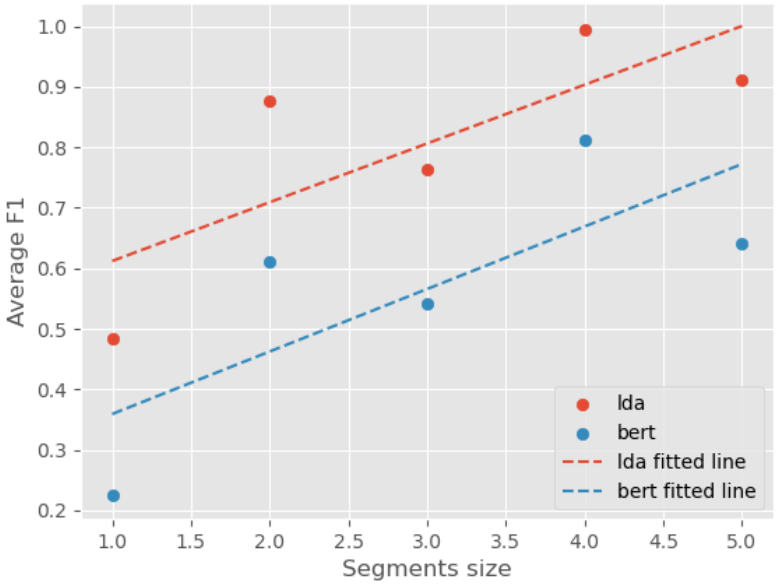

Figure 3.7: Average F1 score for LDA and Bertopic given different segment sizes of the ACL dataset. The overall average F1 score for LDA is 80.1%, and for Bertopic is 56.6%.

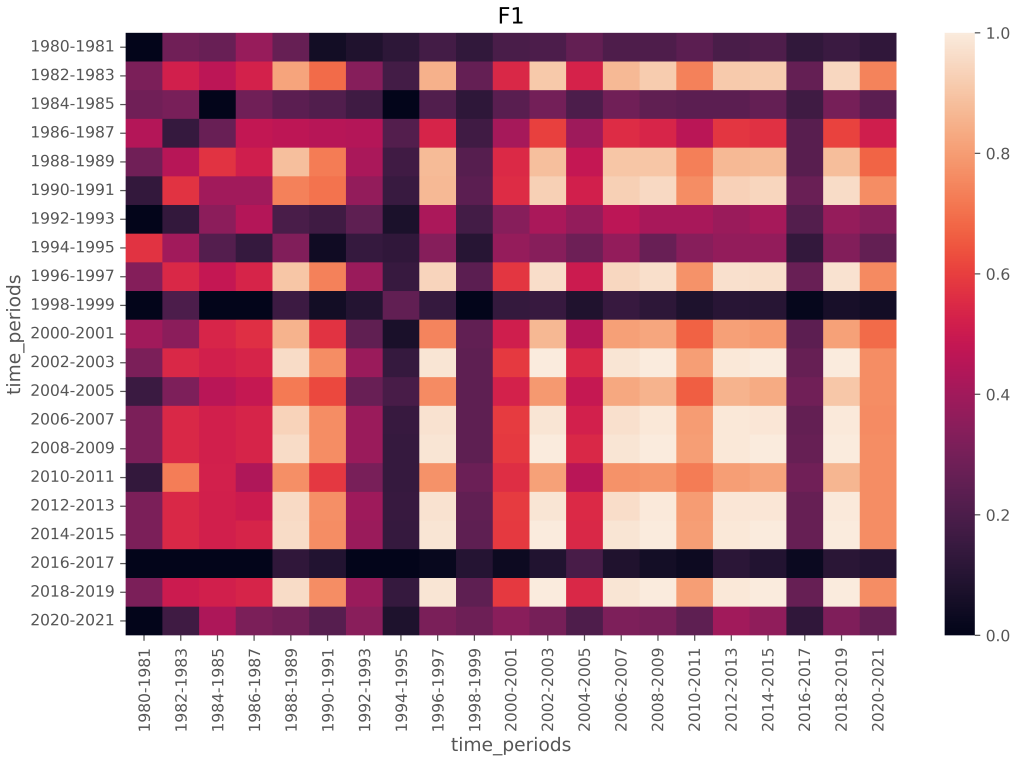

Figure 3.8: Heatmap of F1 scores for LDA on segment size of 2. For each cell, the model is trained on the training data from the row time periods (2-year span) and tested on the test data from the column time periods (2-year span).

3.6 DISCUSSION

Our findings reveal several key insights regarding matching strategies, dataset differences, model performance, and topic emergence detection effectiveness.

COMPARISON OF TOPIC MATCHING STRATEGIES Our results show that TWOR consistently retrieves more semantically coherent topics across datasets, while KL divergence prioritizes topics with similar temporal trends rather than direct word overlap. This is evident in the WoS dataset, where TWOR identifies a topic closely associated with Diabetes, whereas KL divergence retrieves bioelectromagnetic-related terms. A similar pattern is observed in the ACL dataset, where TWOR effectively retrieves topics on Grammatical Error Correction, while KL divergence results in more general methodological terms.

These findings suggest that TWOR is more reliable for direct topic alignment, especially for structured datasets like WoS and ACL, where well-defined topics are expected. However, KL divergence can still be useful for detecting "related but distinct" topics, reflecting indirect relationships in topic evolution [197]. In datasets like Enron, where topics are less structured, KL divergence may capture latent associations between business-related discussions.

Figure 3.9: Heatmaps of F1 scores for LDA (top) and Bertopic(bottom), with the segment size of 3, meaning local models are trained on training data from a 3-year span and tested on test data from a 3-year span.

DATASET DIFFERENCE IN TOPIC EXTRACTION Among the datasets we evaluated, we observed that topic quality varies depending on the level of structure in the text. WoS and ACL datasets yield clearer, more interpretable topics, likely due to the structured nature of research articles, which focus on well-defined subjects. The Enron dataset produces broader, more ambiguous topics, reflecting the informal and unstructured nature of email communications, where multiple themes may coexist in a single document.

These findings highlight the importance of dataset structure when applying topic modeling methods. While structured texts allow for more precise topic

modeling, unstructured datasets may require additional preprocessing or more advanced modeling techniques to improve topic coherence.

COMPARISON OF TOPIC EXTRACTION ACROSS MODELS Our analysis also highlights differences in how CoWords, LDA, and Bertopic extract topics. First, CoWords and Bertopic tend to extract topics with more overlapping words, likely due to their clustering-based approach. Second, LDA generates broader topics, with more diverse top words, whereas Bertopic and CoWords extract more specific topics (e.g., "spelling" and "spell" appearing in the same topic). Third, Bertopic's topics appear more volatile, possibly due to its reliance on contextualized sentence embeddings, which may lead to greater variability in topic representations over time.

These findings suggest that LDA provides a more generalizable representation of topics, while Bertopic and CoWords tend to extract more granular topics with repeated terms. This may explain why LDA aligns more closely with global trends, whereas Bertopic exhibits more sensitivity to transient or highly specific topics.

MODEL PERFORMANCE ON TOPIC EMERGENCE DETECTION Given our qualitative analysis, we observe that CoWords performs better at detecting topic emergence than LDA and Bertopic. However, due to its inherent limitation on generating document-topic distribution, the quantitative metric is not directly applicable. Therefore the limitation of this conclusion is that it is based on limited qualitative results, and might be limited to generalize to a larger scale. On the other hand, evaluated by our proposed quantitative metric, we have a few observations. First, LDA detects emerging topics earlier and more consistently than Bertopic, supporting prior findings that probabilistic topic models perform well even with smaller datasets. Second, Bertopic's performance improves with larger time spans, likely due to its reliance on sentence embeddings, which require more data to generalize effectively.

While CoWords appears to detect emerging topics earlier, its limitations must be considered. Unlike LDA and Bertopic, CoWords does not produce document-topic distributions, which prevents its evaluation using quantitative metrics. Another limitation is that CoWords relies purely on word co-occurrence, lacking the semantic depth of LDA's probabilistic modeling or Bertopic's transformer-based embeddings. As a result, topics extracted by CoWords may be more fragmented or overlap significantly, particularly in datasets with subtle conceptual distinctions. In contrast, LDA and Bertopic provide more structured and interpretable topic representations, albeit at the cost of slower emergence detection.

IMPACT OF DISRUPTIVE PERIODS ON EMERGING TOPICS DETECTION Notably, the lower F1 scores in the 1980s and post-2015 suggest that emerging

topics were harder to detect in these periods. This aligns with historical trends: early NLP research was sparse before the 1990s, while post-2015 saw rapid shifts in deep learning techniques. Our heatmap analysis further indicates that local models struggle to detect emerging topics when disruptive changes occur, a finding consistent with prior work on language model generalization across time.

LACK OF COMPARATIVE EVALUATION IN PRIOR WORK Evaluating topic models is inherently challenging, as traditional metrics like coherence and perplexity measure topic quality but fail to capture a model's ability to detect emerging topics. Prior work on emergence detection, such as BERTrend [17] and ATEM [168], introduces novel methods for emerging trends detection. However, these studies focus on single-model evaluations rather than comparative benchmarking. Most studies assume that a single model is sufficient, overlooking the differences in how various models detect emergence over time. Our study highlights that CoWords detects emergence earliest, LDA provides stable trends, and Bertopic captures nuanced contextual shifts but exhibits higher volatility. These comparative insights are crucial because existing methods often rely on external metadata (e.g., citations or manually labeled emerging topics) rather than evaluating topic models independently. By proposing a self-contained evaluation metric, we provide a more robust framework for assessing when and how different models detect topic emergence, contributing towards more reliable evaluation strategies.

3.7 CONCLUSIONS

In this study, we systematically evaluated three topic modeling approaches, CoWords, LDA, and Bertopic, for their ability to detect emerging topics across structured (WoS, ACL) and unstructured (Enron) datasets. Our analysis focused on topic matching strategies, differences in topic extraction quality, and an independent evaluation metric for assessing topic emergence detection.

Comparing two topic matching strategies, our findings indicate that TWOR consistently produces more semantically coherent topic matches than KL divergence, particularly in structured datasets such as WoS and ACL, where well-defined topics facilitate more precise alignment. KL divergence, on the other hand, tends to retrieve topics with similar temporal trends rather than direct word overlap, making it useful for identifying related but distinct topics. These results highlight the importance of choosing an appropriate matching strategy depending on the dataset structure and the intended analysis goal.

When evaluating on different datasets, we also observed notable differences in topic extraction quality across datasets. WoS and ACL datasets produced more interpretable topics, while the Enron dataset resulted in broader and more ambiguous topic clusters. This difference suggests that the effectiveness of topic

models is influenced by the structure of the underlying corpus, with more formal and domain-specific texts yielding clearer topic representations.

When evaluating model performance in detecting emerging topics, our proposed metric showed that LDA consistently outperformed Bertopic in terms of agreement between local and global models. LDA achieved a higher average F1 score across all segment sizes, indicating its robustness in detecting emerging topics using limited data from specific time periods. Bertopic exhibited greater sensitivity to novel topics but showed lower overall consistency, which may be attributed to its reliance on sentence embeddings that require larger datasets to generalize effectively.

3.7.1 *Limitation and Future Work*

Despite these insights, our study has some limitations. First, we evaluated topic emergence in a retrospective setting, meaning that our approach does not predict future topics but rather identifies when topics gain prominence over time. Second, while we examined three widely used topic models, future work could explore additional topic models to determine whether they provide advantages in emergence detection. Finally, our analysis focused on specific domain datasets (scientific papers, emails), and extending the study to other domains, such as news articles or social media data, could provide further validation of our findings.

In future work, we aim to expand our study to incorporate additional topic modeling approaches from the three categories, such as algebraic topic model NMF(Non-Negative Matrix Factorization) [239] instead of CoWords, probabilistic topic model HDP(Hierarchical Dirichlet Process) [194] instead of LDA, as well as neural topic model ETM (Embedded Topic Model) [55] instead of Bertopic. These models could provide deeper insights into semantic topic shifts and representation learning for emergence detection.

Additionally, we plan to extend our analysis to datasets from diverse domains, including finance, where emerging topics can influence market trends, and news or online communication, which often exhibit rapid topic evolution and linguistic variability. Exploring different textual styles, such as formal reports versus social media discussions, will help assess the adaptability of topic models across different communication formats.

These results show that while neural topic models can capture broad thematic structures, they still fall short in reliably detecting topics that are dynamic and ever-changing. Since topics can be treated as soft, latent concepts over text, this limitation matters for evolving knowledge graphs, which often require timely concepts, entities, and relations. If these neural methods cannot reliably extract new latent concepts, this can translate to a challenge in constructing knowledge graphs that evolve.

After identifying this second challenge, we then move on to the third challenge: long-form multi-turn documents for cross-document coreference resolution in the next chapter.

4

THE CHALLENGES OF CROSS-DOCUMENT COREFERENCE RESOLUTION FOR EMAIL

In this chapter, we focus on the long-form, multi-turn nature of documents, particularly in the context of cross-document coreference resolution (CDCR). Unlike standard coreference resolution, which is typically confined to single documents, CDCR requires models to resolve entity mentions across multiple documents or conversational turns, making it especially challenging in informal, evolving domains like email threads.

We formulate our research question as follows:

RQ3 *What are the challenges for Pretrained-then-Finetuned models in cross-document coreference resolution for complex conversations?*

To address this, we evaluate BERT-based models fine-tuned for coreference resolution and test them on multi-turn email conversations. These emails often contain ambiguous pronouns, informal language, and sparse context, features that severely stress the limitations of task-specific models in long-form, real-world scenarios.

4.1 INTRODUCTION

Coreference resolution (CR), the task of determining which textual mentions refers to the same entity, is a long standing and important task in natural language understanding (NLU) [96]. Being able to perform coreference resolution is particularly important in knowledge capture settings dealing with emergent entities or entities which are not represented in existing knowledge graphs [138, 178].

One such setting is knowledge capture from long form conversations such as medical conversations [209], personal dialogues [171], threaded discussion forums [228], and email conversations [49]. CR in conversational data is challenging because of the change in speakers and high lexical ambiguity [228]. Such lexical ambiguity is caused by the often large amounts of shared background knowledge needed to understand the conversation [191]. For example, imagine an email thread discussing the edits to a document. The participants know the contents of the document and its intended purpose, but such knowledge is not easily available to an extraction system.

In this chapter, we study CR performance on a type of long form conversation, namely, email conversations. Email provides a challenging but realistic case because it contains more than one speaker, and has no limitation on the

length of the emails. Surprisingly, there has not been a lot of work on CR in emails, and only recent work has begun to study CR on emails using deep learning models [47] based on a small hand-annotated subset of the Enron email corpus [101]. The corpus only contains coreference annotations for mentions within email and across emails in each thread. Dakle, Desai, and Moldovan [47] formulate this problem as Within-Document Coreference Resolution (WD-CR), treating an email thread as a single document, and ran a WD-CR state of the art method, spanBert [92]. Given that entities are often referred to across emails and between multiple threads, we instead consider a cross-document formulation more natural. Therefore, we evaluated a state of the art cross-document coreference resolution (CD-CR) [25] model on the same corpus, considering each email in a thread as a single document.

Our results show that email conversations are a particularly challenging domain, where the state-of-the-art performs significantly worse than when applied to commonly-used corpora in the field such as news (e.g. CoNLL F1 34.4 on news vs. 27.4 on email). Based on these results, we outline the key challenges for CR in emails and paths forward to addressing them. In summary, the contributions of the paper are as follows:

1. performance results for the state-of-the-art CD-CR on email conversations including an ablation study investigating performance with and without pronominal coreference;

2. a qualitative error analysis that identifies where the challenges in this domain arise from; and

3. paths forward to addressing this important domain.

4.2 BACKGROUND

We briefly introduce the state of the art on coreference resolution (CR), focusing on the cross-document setting and email. We refer the reader to [63, 191] for recent reviews of approaches to CR.

State-of-the-art CR: In the early literature, CR systems typically contained two separate stages: mention extraction with existing feature-engineered systems (e.g. syntactic parsing) and coreference relation finding. The first work that joined the two stages was e2e-coref [110]. It proposed an end-to-end deep learning architecture that jointly learned to extract mentions and rank them as to whether they corefer. Later, c2f-coref [111] proposed a span refinement technique to iteratively refine the span representations of mentions to help address global inconsistency and achieve higher-order inference. The current state of the art in within-document CR, spanBert [92], leverages large pre-trained language models [93].

However, the success of such models relies on textual orders. In the cross-document (CD) setting, generally there is no temporal ordering between the documents, which impacts negatively the performance of these models [25]. Instead of learning an antecedent distribution and chaining the current mention to the most probable antecedent, the current state-of-the-art [25], which we evaluate in this chapter, learns to extract candidate mentions and compare all the most probable mentions for potential coreference. We explain the model in more detail in Section 4.3.

To evaluate the performance of CD-CR, ECB+ (EventsCorefBank+) [45] is the most commonly used dataset in recent years. It consists of news articles organized by topics, and contains both WD and CD CR annotations for both entities and *events*. To address the lexical ambiguity challenge in CD-CR in particular, within each topic, each instance (also called *subtopic*) is a pair of similar but different events. ECB+ is a challenging dataset that focuses mostly on the specific situation for disambiguation for events that contain similar entities. In the rest of the paper, we show that CD-CR models that perform reasonably on ECB+ do not work well on email.

More broadly, Cattan et al. [25] highlighted the pressing need for more realistic evaluation on CD-CR task. In particular, existing work often reports performance using golden mentions instead of predicted mentions. Their work showed that the contribution of mention prediction is significant and should be considered in evaluation. Thus, an end-to-end training approach is more realistic for real-world data.

CR for email: Email conversations have long been studied on tasks such as classification, search and summarization [47]. One of the largest email corpus is the Enron Email Corpus [101] containing emails of 150 employees of the Enron Corporation. Surprisingly, only recently has CR within email come to the fore. In particular, [47] provided, what appears to be, the first analysis of entity coreference for email in the literature. They introduced a manually annotated seed corpus (SC) containing 46 threads and 245 emails from the Enron Email Corpus. The authors filtered the original corpus so that each thread contains more than 3 emails in order to have both within-email annotations as well as cross-email annotations, and each email has meaningful text body instead of forwarding messages. In their analysis they evaluated a state of the art model, spanBert [92], that they had fine-tuned to the seed corpus, which had 54 F1 score. This is 26 points below the state-of-the-art in general WD-CR [224].

Building of this work, Dakle and Moldovan [49] introduced a larger dataset called CEREC with 6001 email threads containing 36,448 emails with weakly labeled data. The labeling has two stages, mention identification annotations and coreference relation annotations. Both stages use pre-trained spanBert [92] to annotate without further training. The mention annotations produced by spanBert are then manually corrected. For the coreference relations annotation, first a small subset of email threads are manually annotated as a validation set

for training performance. Then spanBert is trained on the manually annotated seed corpus [47] and then the coreference relations are obtained on the large dataset based on the golden mentions. Surprisingly, training a state-of-the-art WD-CR model on CEREC showed roughly no change in performance, with a reported F1 score of 54.1 [49] compared to the F1 score of 54 [47] on the seed corpus.

4.3 METHOD

We now describe the model and model training we use to characterize the performance of CD-CR in email.

Model architecture: As previously discussed, we use the state-of-the-art model architecture [25] and summarize it here:

The model contains three main modules: a `span_scorer`, a `span_embedder`, and a `pairwise_scorer`.

The model takes a set of documents as input and uses a pre-trained language model to get a contextual representation of each token in the document. It then segments the tokens to determine possible mention candidates up to a pre-defined mention width. Then, the `span_embedder` is used to obtain the embedding for each mention candidate. The model then prunes all candidates according to `span_scorer`. The most probable candidates are paired together and scored by the `pairwise_scorer`, which is a multi-layer perceptron (MLP). During inference, agglomerative clustering is used to return final clusters of coreferent mentions.

Cattan et al. [25] proposed three different training styles: pipeline, continue and end-to-end (e2e). The main difference is that they freeze different parts of the pipeline. We use the e2e style, which trains all three modules end-to-end. A binary cross-entropy loss is used to jointly train the `span_scorer` and `pairwise_scorer`.

Model training: We train this model on the SC corpus [47] introduced previously. We use the e2e training style to achieve a more realistic evaluation. For comparison to ECB+ results, we assume that each email thread is equivalent to an ECB+ topic. We split long documents to the maximum document length of $n = 512$ tokens.

We use a span representation based on e2e-coref [110]. First, we use pre-trained RoBERTa [238] to encode each token in the input. For each mention candidate, or *span*, i we compute the embedding s_{emb} as a concatenation of 4 different components:

$$s_{emb}(i) = (x_{start(i)}, x_{end(i)}, \hat{x}_i, \phi(i))$$

where $x_{start(i)}$ and $x_{end(i)}$ are the token representation of the first and last token in m_i,while $\hat{x}_i$ is the weighted sum of all token representations in span i (i.e. the

attention), and $\phi(i)$ is the feature vector that encodes the length information of span i.

Then all encoded mentions will be scored by the mention scorer $s_m(\cdot)$, which are then pruned to retain only $\lambda = 35\%$ percent of mentions. We use a multiple layer perceptron (MLP) layer with ReLU activation function as our $s_m(\cdot)$. Then the pruned mention candidates will be paired up and scored by a pairwise scorer $s_p(i,j)$, where i,j might or might not from different documents. The pairwise scorer $s_p(i,j)$ is also a MLP. We perform negative sampling.

All three modules mention scorer $s_m(\cdot)$, span embedder s_{emb} and pairwise scorer $s_p(i,j)$ are jointly trained by optimizing the binary cross-entropy loss over pairs:

$$L = -\frac{1}{N}\sum_{i,j\in N} y \cdot \log(s(i,j))$$
$$s(i,j) = s_m(i) + s_m(j) + s_p(i,j)$$

where N is the set of mention pairs and y indicates the binary label. When $y = 1$, it indicates the mention pair is coreferent.

Experimental settings The experiments are carried out on titan RTX 24GB GPU, it takes approximately 70 minutes to train and evaluate 10 epochs. Our training data is split into training, validation and test set in an 80:10:10 ratio. The seed corpus (SC) contains 43 email threads and 228 emails in total. All threads have at least 4 emails. The whole dataset contains 3815 mentions across emails within threads. Each email contains header, body and footer.

We note that emails have conversational features, i.e. the speaker of each email changes in a thread and therefore it is more confusing to learn the antecedent distributions for pronominal mentions across emails. Simple rules for *{I, you}* are easy to resolve, but for second order pronouns (e.g. our team) and above it is challenging to learn the distribution.

To further study cross-documents pronominal resolution, we first remove first order pronouns in a union of {I, you}, and then remove a whole list of pronouns [1] from the dataset. Table 4.1, details the number of removed mentions.

Mention type	Number	Subtracted
All mentions	3815	0
- subset of pronouns {i, you}	3298	517
- all pronouns	2613	1202

Table 4.1: Number of mentions before & after removing pronouns.

1 The full list of pronouns that we removed is here: https://gist.github.com/effyli/da7c4243f296a6c689697384b48896f5

	MUC			B^3			$CEAF_e$			LEA			CoNLL
Dataset / Setting	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1	F_1
ECB+ Entities Test set	41.7	52.3	46.4	24.8	37.1	29.7	27.4	26.8	27.1	22.3	34.4	27.1	**34.4**
Email Test data	41.2	25.0	31.1	40.4	22.5	28.9	37.1	15.9	22.3	26.3	12.0	16.5	**27.4**
Subset pronouns {I, you}	41.7	24.7	30.9	40.1	23.7	29.9	31.3	14.0	19.3	20.6	11.7	14.9	26.7
All pronouns	34.8	30.0	32.2	21.0	37.6	27.0	28.5	6.8	11.0	10.0	18.0	12.9	23.5

Table 4.2: Cross-document coreference results. ECB+ Test Set with entities only on the topic level as a baseline (from [25]). Email test set [47], with/without subset {I, you} or removing all pronouns.

4.4 RESULTS

We evaluate the model on the standard coreference resolution metrics, including MUC, B^3, CEAFe, LEA. The results are reported in Table 4.2. We compare our cross-document coreference resolution model on the test set of the email seed corpus with the ECB+ test set. To make the results comparable, we compare the evaluation result on the ECB+ test set for entity resolution only, with predicted mentions, and on a topic level. The final F1 score is 34.4. For the email seed corpus, we first evaluate our model on the full test set with all pronouns. The F1 score is around 27.4, which is a significant 7-point drop. Then, a subset of pronouns {I, you} is removed from the mentions. This has a slightly worse F1 score of 26.7. After the removal of all pronouns, the model produced an F1 score of 23.5.

This shows that the model is able to learn some alignment between pronouns but performs worse on less generic mentions, which often characterize entities.

4.5 CHALLENGES

We now discuss the main challenges faced in order to improve coreference resolution in email using a qualitative error analysis. Subsequently, we articulate several directions for future work.

Informal language: As mentioned in Section 4.2, most datasets for the CD-CR task are in the *news* domain, where the language style is more formal and structured. Hence, most entity mentions have previous references. In comparison, email is more informal and hence less structured. Table 4.3 shows examples of this from the ECB+ and Email datasets. In the ECB+ example, we can see that the text is clear. The main coreference challenge is that two different events have similar names. In addition, in the Email example, the coreference relations are more complicated. In the Document 1, one would need header information to reason what *we* refers to. Similarly, the two emails are needed jointly to understand the coreference for *your* in Document 2 which must exclude the

speaker herself. This example illustrates that treating coreference resolution in emails as within document is insufficient.

Variety of surface forms: Table 4.1 shows pronoun-related mentions take up to 31.5% of all mentions. As discussed in Section 4.4, model performance drops by 4 points on the F1 score after removing all pronouns. This indicates that the model struggles on predicting coreference between other type of mentions. Some example mentions are shown in Example 1.

Example 1

```
'Frazier,Perry'
'Perry'
'FP'
'perry.frazier@enron.com'
```

From the example, we can see that there are multiple different surface forms for the same entity and these forms vary widely. Models need to become better at coping with this sort of wide variation. This is a known challenge in the literature [191] but given the nature of email appears with more frequency.

Sparsity: The prior two challenges are exasperated by the fact that there is a lack of data necessary to train good models. Unfortunately, the current weakly labeled CEREC dataset is inadequate due to low quality annotations.

Dataset	Doc 1	Doc 2
ECB+	News that Barack Obama may name Dr. Sanjay Gupta of Emory University and CNN as his Surgeon General has caused a spasm of celebrity reporting.	President Obama will name Dr. Regina Benjamin as U.S. Surgeon General in a Rose Garden announce ment late this morning.
Email	Audrey, how about moving the meeting to 8:30? We will have to leave here by 9:35 or so to get a seat at the employee meeting. Kim	Okay, let ' s move Steve ' s Strategy Meeting to 8 : 30a on the 23rd. Please adjust your calendars accordingly. adr Audrey D. Robertson

Table 4.3: Examples from ECB+ and SC (emails). The same color denotes coreference. Emails are from the same thread.

4.6 PATHS FORWARD

More data: A clear path forward is the provision of more annotated data. Here, we suggest that instead of using an existing coreference model to generate data as in CEREC, a data programming approach [172] might be more appropriate.

Incorporation of rules: While, as we have discussed, email conversations are complex, there are opportunities to take advantage of common patterns within conversations. For example, we calculated that using simple rules to align subset of pronouns {I, you} with email headers could resolve around 13.6% of mentions. Incorporating these and other rules with current models is a promising direction.

Language models for email text: Creating a pre-trained model specifically for emails could help to better capture the unique idiosyncrasies of email. This could also address the huge amount of memory needed for token encoding.

Better pruning strategies: The current state-of-the-art model on CD-CR, currently pairs up all mention candidates. This creates a massive search space thus requires a pruning factor to be given beforehand. Dynamically pruning candidates with a smarter strategy could reduce this space of potential candidates.

Improving span representations: Current span representations used in these models lack contextual information. Such contextual information is important for email (e.g. what conversation an email is occurring in). Refining span representations by incorporating whole document contexts or speaker information is an important direction forward.

Word Knowledge & Reasoning: Lastly, one common challenge in current NLU system is the lack of world knowledge. This also holds true for coreference resolution [191]. General information that most parties in the conversation or audiences would recognize will be missing in the email itself. Thus, it may be the case that, in particular for long form conversations, a background knowledge base is a prerequisite for good performance. Such a knowledge base might not be in the form of a knowledge graph but could be in the form of background documents.

4.7 CONCLUSION

In this chapter, we investigated the performance of the state-of-the-art for cross-document coreference resolution on email. We have shown that email is a challenging domain for existing pretrain-then-finetune models, because of their long-form, multi-turn, and informal language usage characteristics. Based on these results and a qualitative analysis, we have identified six paths forward to improve performance in this context.

This concludes the three challenges we identified with the Pretrain-then-finetune paradigm for knowledge graph construction tasks. In the next part of the thesis, we shift our focus toward solutions that are aided by the Pretrain, Prompt, and Predict paradigm with large language models.

Part II: Designing Pretrain, Prompt, Predict Paradigm-based Workflows for Knowledge Graph Construction.

SUMMARY OF PART II

In Part II of this thesis, we shift focus to the Pretrain, Prompt, and Predict (PPP) paradigm for Knowledge Graph Construction. Leveraging the in-context learning capabilities of large language models (LLMs), we demonstrate how prompting strategies and instruction tuning can enable generalization across tasks and domains, without the need for task-specific training. While this paradigm addresses many challenges identified in Part I, it also introduces new ones, particularly around cost-efficiency and orchestration. To this end, we propose methods that improve the usability of PPP-based pipelines through code-generation workflows, offering scalable and flexible solutions.

5

KNOWLEDGE-CENTRIC PROMPT COMPOSITION FOR KNOWLEDGE BASE CONSTRUCTION FROM PRE-TRAINED LANGUAGE MODELS

In this chapter, we provide a holistic evaluation of using the Pretrain, Prompt, Predict (PPP) paradigm for Knowledge Graph Construction (KGC), demonstrating the potential of prompting large language models (LLMs) for end-to-end knowledge base construction. Specifically, we investigate whether LLMs can accurately complete triples by predicting the object given a subject and a relation. We explore how to construct effective prompts, select in-context examples, and enhance factuality through external knowledge sources such as Wikidata.

5.1 INTRODUCTION

The field of Artificial Intelligence (AI) has seen huge improvements in tasks related to language due to Pre-trained Language Models (PLMs)[116] and the computational efficiency introduced by transformers [200]. This significant improvement can be seen in areas such as translation, summarisation, and classification [9, 37, 154].

Given their effectiveness in many information extraction tasks [9, 87], there has been a movement by the community to study their use in tasks focused specifically on knowledge base construction [16, 140]. As part of that larger interest, The Knowledge Base Construction from Pre-trained Language Model Challenge (LM-KBC) was launched in 2022 to better understand the role that PLMs can play as a source of knowledge themselves [188]. Essentially, providing a framework to study how one can construct a knowledge base directly from a PLM.

This report presents our approach and results for the second edition of the LM-KBC challenge at the 22nd International Semantic Web Conference (ISWC 2023). The challenge is to predict objects for a given subject-predicate pair. An example is given a subject, *Matt Damon*, and the relationship we are targeting, *person has number of children*, retrieve the object, in this case, a number, for that pair. It may be that the model needs to predict an existing Wikidata object, example being, the subject country *Fiji*, has associated geographical states, and the objects we wish to retrieve are those Wikidata states.

We propose a pipeline for knowledge base construction by prompting large language models, specifically, GPT-3.5 and GPT-4. We explore different setups with in-context learning by utilizing an example selector and knowledge-enriched

prompts to provide more contextually relevant prompts. Our results show rule-based example selectors considering cardinality per relation exhibit significant performance on the task. Furthermore, enriching entities and relations with additional properties obtained from GPT-4 help boost the performance even further.

5.2 RELATED WORK

The notion of using a language model as a source of knowledge itself was brought to the fore by the LAMA paper in 2019 [161]. This can be seen as one part of the larger move towards prompting PLMs to solve NLP tasks. We refer the reader to the survey by Min et al. [145] for a deeper dive into prompting and associated architectures for NLP. Here, we focus on work directly related to the LM-KBC challenge. An overview of the various approaches can be found in the 2022 challenge introduction [188].

Specifically, our work follows on from the winner of task 2 of last year's challenge, "Prompting as Probing"[2]. In their work, they prompt GPT3 with manually curated prompt templates, including 4 examples from the training set in their prompts. These are then updated with the specific subject entity of interest during the prompting workflow. Additionally, they include a post-processing step called "fact-probing" in which the PLM is asked to judge whether a given result produced by PLM is indeed true. This helps improve the precision of the model. The authors went on to perform an ablation study outside of the challenge whereby they utilised Wikidata to help improve entity disambiguation during the post-processing step. By using Wikidata information, such as the hypothesised concepts type in relation to the relationship used during prediction, to validate the prediction. This study proved an slight performance gains, however this was not allowed to be part of the reporting in 2022, but in 2023, such retrieval augmentation is allowed. We employ a similar approach here.

Our approach differs because we focus on dynamically selecting examples from the training set to include in the prompt. Additionally, our prompts provide more context than those used by Alivanistos et al [2]. Also, we note that we use a newer version of GPT.

In [143], a benchmark is provided to establish the ability of models to construct knowledge graphs from text. The authors provide an ontology description as part of their prompts. The prompt consistently employs the *relation(subject, object*) format to represent relationships and expects the model's output to adhere to this notation. We also employ ontology descriptions (i.e. extra knowledge base context) in our prompts.

5.3 LM-KBC CHALLENGE DEFINITION

The Language Model Knowledge Base Construction (LM-KBC) challenge task is defined as follows. Take a set of subject (s) predicate (p) pairs ($< s, p >$) and predicting a set of objects (o_1, o_2...) in relation to those pairs. The target set of objects can be; (1) a wikidata identifier, (2) a numerical value, or (3) empty.

The LM-KBC Challenge provides two distinct tracks for participants. The first, known as the "Small-model Track," restricts participants from using pre-trained Language Models with no more than 1 billion parameters and excludes the use of contextual information. The second termed the "Open Track," imposes no limitations on the model size and permits the inclusion of contextual data. For the purposes of our research, we tackle the Open Track.

5.3.1 *Dataset*

The dataset available for LM-KBC comprises 5820 samples (i.e. triples) evenly divided over training, validation and test set. The objects within the test set were withheld during the time period when our system was developed. The dataset encompasses an array of 21 distinct relations, where a diverse range of subject-entities are provided for each relation. Each triple contains both the Wikidata identifiers and also lexicalizations of each element of the triple as English text.

Each relation in the train and validation sets is accompanied by a set of ground truth object-entities, curated to align with specific subject-relation pairs. It is noteworthy that the length of object-entities affiliated with a given subject-relation pairing exhibits variability. Meaning, in the test set, the implementation must correctly predict sets of objects and empty sets.

We note that there are four relations, e.g. *PersonHasPlaceOfDeath*, where there are potentially zero relations to the subjects in the available sets. In comparison, *CountryHasStates* requires from between one and twenty objects for the prediction of the related subjects. Furthermore, objects are not limited to other Wikidata entries, the entries could also be numerical, e.g. *SeriesHasNumberOfEpisodes*.

5.4 METHODOLOGY

In-context learning is a fundamental capability in many language modelling approaches. The approach is prominent in GPT models particularly starting from GPT-3 [18]. Our approach centers around in-context learning (i.e. prompting, few-shot learning), which combines the capabilities of pre-trained language models with the contextual information available in the text [133]. Specifically, we focus on the design and utilization of few-shot prompts. Prompting refers

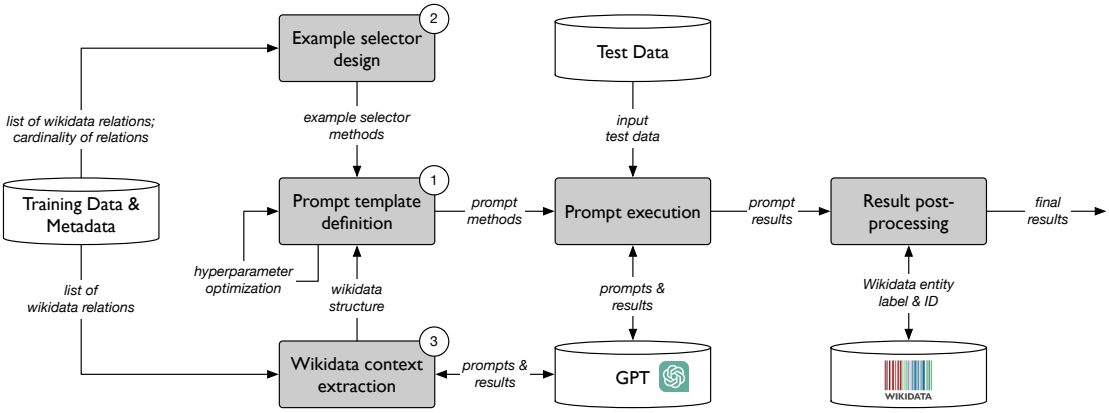

Figure 5.1: An overview of the *thames*' team method.

to the use of specific instructions and/or statements to induce the model to complete certain tasks. Few-shot learning is an approach where the language model learns how to perform a task from minimal data points, i.e., learning the task from only a few examples (few shots). Considering prompting and few-shot learning principles, we carefully designed our prompts that integrate both static and dynamic elements. An overview of our method is provided in Figure 5.1. From the training dataset, we derive a list of wikidata relations, along with their cardinality distribution. We use the set of relations for two purposes: example selector design and Wikidata context extraction that is prompted from GPT models. The prompt template is subsequently defined by these two components, followed by a refining process for hyperparameters such as the number of examples selected based on the final model performance on the evaluation set. We then execute the prompts with test data through GPT models. Finally, the generations are post-processed and connected to Wikidata IDs. We will explain the components of our approach in the following sub-sections.

5.4.1 *Prompt Template Definition*

In compiling our prompts, we employ static prefix and suffix components while dynamically selecting examples in between based on the given subject-predicate pair. We incorporate static elements at the beginning and end of each prompt to provide a consistent context for the language model. The prefix scopes the prompt, guiding the model to understand the relevant parameter space, while the suffix ensures structure and uniformity. The crux of our methodology lies in selecting and integrating dynamic examples from the training dataset into few-shot prompts. This process is facilitated by the use of two distinct example selectors, designed to guide the language model's comprehension of the extraction task at hand.

GPT-3.5 and GPT-4 have been fine-tuned utilizing dialogue and instruction datasets [155]. On top of this finetuning, they are both optimized for dialogue

and instruction followed by using Reinforcement Learning with Human Feedback (RLHF) [154]. RLHF is a machine-learning approach that involves mapping out optimal strategies based on human responses. This technique allows the language model to learn more complex behaviours and concepts that are difficult to define or specify explicitly in a traditional reinforcement learning setup. By incorporating humans in training, the model can inherit a more nuanced understanding of several tasks [38]. Leveraging this background knowledge, we carefully phrase the static components of the prompts.

Initially, the prefix assigns a role to the LLM, i.e. "Act as a knowledge base", "Imagine that you are Wikidata", etc. Then, a brief task description is given, followed by an explicit statement indicating that the prompt will continue with examples. A fixed example template has been devised to be populated by the example selectors. The suffix then delineates the conclusion of the selected examples, stating that it is now the LLM's turn for prediction. The prompt ends with a template to be filled with the input subject-predicate pair and a signal of continuation, i.e. `":"`, `"["`, etc.

5.4.2 *Example Selectors for In-Context Learning*

Context sensitivity is a recognized phenomenon in in-context learning [18]. The immediate textual content that appears prior to the prediction point is the sole form of the input. Everything the model generates from that point is a continuation of the prompted input. This sensitivity can be both beneficial and problematic. An advantage is that it enables the model to adapt rapidly to changing task requirements and examples. As a drawback, the high sensitivity can lead to issues with model consistency and predictability, resulting in a hallucinatory generation. Therefore, prompts play a crucial role when it comes to extracting knowledge from the language model.

We account for context sensitivity in the selection of both static and dynamic components of the prompt. However, the dynamic selection of the most relevant examples is specifically designed to leverage context sensitivity. Our example selectors pick out the most relevant instances from the training set. The rule-based selector follows certain rules for the selection while the similarity-based selector leverages cosine similarity. The selectors are detailed in the following subsections.

5.4.2.1 *Rule-based Example Selector*

The rule-based example selector is designed as a systematic approach to sample examples from the training set. Given that instances may have zero or more objects, this approach ensures that the diverse nature of examples is taken into account. To instil this understanding, we enriched our prompts with five spe-

cific examples for each instance. The selection criteria for these five examples are as follows:

- Minimum Object Example: We selected one example with the fewest number of objects for a given relation.
- Maximum Object Example: We selected one example with the highest number of objects for a given relation.
- Random Selection: To add an element of variability and ensure broader coverage, we incorporated three additional examples. These were chosen at random from the training set.

This strategy helps in achieving a balanced representation of the data, ensuring that the model does not develop a bias towards any particular pattern.

5.4.2.2 *Similarity-based Example Selector*

The similarity-based example selector operates by using semantic similarity measures to identify instances that are akin to the input instance. This approach allows for the dynamic selection of examples that are contextually compatible with the input text. The functioning of this system relies on embeddings and necessitates a list of vectorized or embedded examples, to which the given input can be compared. Furthermore, it computes a semantic similarity score, such as cosine similarity or dot product, in order to select the closest examples from the pool of embedded examples.

As far as performance is concerned when applied to GPT-3.5, this selector is noticeably slower than the rule-based selector, which is expected given its operation at the embedding level. Semantic similarity-based selection methods are more suited to tasks that harbour a high degree of variation and ambiguity. However, the task at hand in this case, shows a lower degree of variation as it is limited to 21 relations.

5.4.3 *Prompt Improvement through Wikidata Context Extraction*

We hypothesise, that given a subject-predicate pair, we can gain greater accuracy when predicting the object if the model is given the correct context for that pair. We extend this further by utilising the schema and knowledge base from which the subject and predicate came from. Specifically for this task, we state that, for a subject-predicate pair from Wikidata, it is possible to use the qualities from that particular knowledge base to enhance the prompt. To this end, we prompt GPT-3.5 to provide a set of relevant contexts related to the given properties. The prompt that we use to extract these contexts is available in our GitHub repository[1]. An excerpt of the result is available in Listing 5.1

1 `https://bit.ly/3QDewyx`

Listing 5.1: An excerpt of the extracted Wikidata context on the given properties from GPT-3.5

```
{
  "CompanyHasParentOrganisation": {
    "value": "P749",
    "wikidata_id": "P749",
    "wikidata_label": "parent organization",
    "domain": "organization",
    "range": "organization",
    "explanation": "This property is used to indicate the parent
       organization
        of a company."
  },
  ...
}
```

An example context usage is provided in Listing 5.2. For this example, the subject, *AT&T*, the Wikidata ID *Q35476* and relation, *CompanyHasParentOrganisation*, are provided by the competition dataset. Using Wikidata context generated through prompt, we are able to enhance the context by finding related information to the given relation. These contexts include subject and object class type, domain and range information, the label of the given relation, and a full description of that label. This context information is injected into the relevant sections of the prompt.

Listing 5.2: An example wikidata context usage within the prompt

```
Your task is to predict objects based on the given subject and relation.
- Given Subject: ('AT&T','Q35476')
- Subject Type: 'organization'
- Object Type: 'organization'
- Relation: 'CompanyHasParentOrganisation'
- Relation Wikidata ID: 'P749'
- Relation Label (Wikidata): 'parent organization'
- Relation Explanation (Wikidata): 'This property is used to indicate
    the parent organization of a company.'
==>
Predicted Objects:
```

5.4.4 *Prompt Execution and Post-processing*

We adapted prompt execution and post-processing parts of the baseline code provided by the challenge. The adaptations are mostly to cater for the needs of debugging and testing. One exception, however, pertains to the post-processing entity with title and subtitle in the results (i.e., results containing character `":"`).

We noticed that some results in the validation set only matched results without the subtitle part. Therefore, we add a slight modification in the Wikidata disambiguation to check for only the main title in case of full string did not return Wikidata IDs. This update helps to improve the results, especially for the `PersonHasAutobiography` relation. Additionally, with our post-processor, we notice that model tends to generate duplicated results and therefore we added a de-duplication step.

5.5 RESULTS

We summarise the results of our system implementation and experiments in Table 5.1. We then present a more detailed comparison between GPT-3.5 and GPT-4 with the highest performing example selection methodology utilised when prompting each model, this comparison is shown in Table 5.2. Also discussed are the results for zero-object cases, this is where the system should correctly *not* predict an object for some subject-predicate pairs.

5.5.1 *Overview*

In our methodology, we discuss two potential example selection mechanisms, where the selected examples are injected into various prompts. We first performed experiments with GPT-3.5 and the similarity-based selection methodology, we then used our proposed rule-based methodology for both models. From our experiments, we find that a rule-based approach to prompt creation yields greater scores in recall and F1 regardless of the underlying model. The use of GPT-4 in combination with the rule-based approach gave the best results overall for all F1 metrics.

Table 5.1: This table presents the results for each of the presented prompt selection methodologies, and for each model utilized in the experiments. The highlighted block presents the highest score per metric.

Model	**Selector**	**Precision**	**Recall**	**F1**
GPT-3.5	Similarity-based	0.5595	0.6154	0.5484
	Rule-based	0.6105	0.6492	0.5863
GPT-4	Rule-based	**0.7128**	**0.6894**	**0.6744**

5.5.2 *Rule-based prompts*

Given that rule-based prompts offer the best results overall in our experiments, we present a more detailed comparison between GPT-3.5 and GPT-4 where a full breakdown of all predicate scores is available in Table 5.2.

Table 5.2: This table presents a side-by-side comparision between GPT-3.5 and GPT-4. Each relation has a breakdown of its precision, recall and F1 against a respective model. The highlighted block presents the highest score per metric.

	GPT-3.5			**GPT-4**		
Relation	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
BandHasMember	0.4998	0.6186	0.5110	0.5507	0.6408	**0.5628**
CityLocatedAtRiver	0.7200	0.5393	0.5885	0.7700	0.5882	**0.6375**
CompanyHasParentOrganisation	0.3400	0.7350	0.3367	0.6400	0.7900	**0.6367**
CompoundHasParts	0.9073	0.8960	0.8983	0.9667	0.9710	**0.9677**
CountryBordersCountry	0.8815	0.7783	**0.8156**	0.8905	0.7543	0.7898
CountryHasOfficialLanguage	0.6364	0.8756	0.6548	0.9218	0.8244	**0.8474**
CountryHasStates	0.7353	0.7156	0.7124	0.7384	0.7436	**0.7381**
FootballerPlaysPosition	0.5600	0.7383	0.6173	0.6850	0.7167	**0.6897**
PersonCauseOfDeath	0.6400	0.7400	0.6400	0.8000	0.8033	**0.7950**
PersonHasAutobiography	0.3081	0.3150	0.3008	0.3517	0.4350	**0.3742**
PersonHasEmployer	0.4860	0.3265	**0.3675**	0.4600	0.3020	0.3422
PersonHasNoblePrize	0.9500	0.9500	0.9500	0.9900	0.9900	**0.9900**
PersonHasNumberOfChildren	0.5000	0.5000	0.5000	0.6500	0.6500	**0.6500**
PersonHasPlaceOfDeath	0.5000	0.6700	0.5000	0.6700	0.7400	**0.6733**
PersonHasProfession	0.3650	0.3312	0.3230	0.5150	0.4086	**0.4268**
PersonHasSpouse	0.6983	0.7050	**0.6983**	0.7033	0.7150	**0.6983**
PersonPlaysInstrument	0.5903	0.4661	0.4801	0.7700	0.4752	**0.5540**
PersonSpeaksLanguage	0.7622	0.8385	0.7062	0.8485	0.8427	**0.7964**
RiverBasinsCountry	0.7253	0.8279	0.7249	0.8434	0.9336	**0.8542**
SeriesHasNumberOfEpisodes	0.4450	0.5100	0.4667	0.5750	0.5800	**0.5767**
StateBordersState	0.5696	0.5568	0.5197	0.6278	0.5724	**0.5616**
Average	0.6105	0.6492	0.5863	**0.7128**	**0.6894**	**0.6744**

Table 5.2 demonstrates the efficacy of GPT-4 over GPT-3.5. GPT-4 outperforms in 18 of the 21 relations that require predictions. The two relations where GPT-3.5 outperforms are *CountryBordersCountry* and *PersonHasEmployer*, while for one relation *PersonHasSpouse*, the two models are tied. We look to further break down the relations in GPT-4 to identify any patterns that may emerge. The three lowest performing classes are *PersonHasAutobiography*, *PersonHasEmployer*, *PersonHasProfession*. All three of these relations are the subject of type

Person. This pattern of poorer performance on Person related relations is common to both models.

5.5.3 *Zero-object cases*

As discussed in Section 5.3.1, it is possible for one of the given subject-predicate pairs that the target object to be predicted is empty. In Table 5.3, we present the F1 scores in regard to this specific issue.

Table 5.3: This table presents the results of zero-object detection for only the rule-based selection methodology across the two models utilized in the experiments. The highlighted block represents the highest score per metric.

Model	**Precision**	**Recall**	**F1**
GPT-3.5	0.6348	0.6854	0.6591
GPT-4	**0.7037**	**0.8920**	**0.7867**

Overall, GPT-4 is the better-performing model for this specific task when using rule-based example selection for prompt construction.

5.6 DISCUSSION

The findings and observations from our study have shed light on a few perspectives when using PLMs for KBC, including how contexts affect in-context-learning performance, the impact of post-processing and what the limitations of GPT-family models are for this task.

Contextual Relevance in In-Context Learning: In our experiments, we observe that both demonstrated examples and additional knowledge of the entities play a crucial role in enhancing a model's understanding and generation. This aligns with the fundamental idea that a richer context helps produce a more coherent response. To improve contextual relevance, one can select more relevant demonstrations given relations and entities to predict. Additionally, providing extra knowledge for relations and entities can help generate more accurate responses.

Impact of Post-Processing: PLMs do not always follow the given instructions. Due to the fact that PLMs are often fine-tuned with natural question-answering style tasks, the generation of answers often comes as a natural language-style answer. Hence, being able to unify the answers and quantitatively evaluate them is a challenge. Follow from this is that effective post-processing strategies are necessary for the generation of quality results.

Performance Enhancement of GPT-4: Our results corroborate the general consensus that GPT-4 improves performance compared to its predecessors, such as GPT-3.5.

Hallucinations on Relation Types: Although GPT-4 has shown significant ability for predicting the objects given subject and relation, the model still shows signs of hallucination. The model especially struggles with specific types of relations such as *PersonHasProfession* and *PersonHasEmployer*. When allowed to generate multiple answers, the model tends to hallucinate after the first correct answer and generate related professions but not factually correct. This might be improved if the model can fact-check with every answer it produces.

Temporal misalignment between Wikidata and GPT-family Models: The dataset from the organizers is from Wikidata, which contains up-to-date knowledge, while GPT-family models were only trained with text till September 2021, resulting in a performance bottleneck due to the nature of the dataset.

5.7 CONCLUSION & FUTURE WORK

We proposed a PLM-based pipeline centered on in-context learning for performing knowledge base construction, specifically, for the task of predicting objects given a subject and relation. We explored different approaches to prompting including the use of contextual information from training and an associated knowledge graph. Our results indicate that providing examples with higher contextual relevance, including the type of relations, and the possible cardinality of the objects, can help with knowledge base construction.

Our results show that PLMs have great potential to perform KBC tasks when prompted effectively. However, we still observe a list of limitations during the process: (1) The temporal information gap within the GPT-family of models may result in providing inaccurate responses. (2) The free-form of generation of generative PLMs makes the evaluation of the model's true capacity challenging. (3) Models struggle with the actual number of answers for relations such as `"PersonPlaysInstrument"` and potentially will hallucinate by returning answers that should not be returned. (4) We require humans to design the prompt template and hence will need to re-design when adapting to other tasks. In the future, different paths can follow to address current limitations.

1. Utilizing automatic prompt optimization techniques such as in [186]. Instead of human modifying prompts, we can learn the most optimal prompts automatically.

2. Chaining large language model prompts [222] to iteratively feed the output of the previous response to the next, aiming to amplify the advantages at each step and provide a more structured interaction with the model. Given this technique, we might be able to address the hallucination problem to some extent. A chain-of-thought prompt provides internal validation and improves the models' robustness in responses.

3. Exploring the effects of example selectors with different attributes. Currently, the example selector only considers the types of relations and the possible number of objects. Another avenue to explore is to select examples based on the properties of each relation type. Being able to understand how exactly example selectors affect the response could help to generalize to other tasks.

The results in this chapter highlight the strength of PPP-based solutions in Knowledge Graph Construction, especially in comparison to PTFT-based models like BERT. The ability of LLMs to generalize across tasks and domains significantly reduces the need for task-specific fine-tuning and labeled data. However, this flexibility comes at a cost: the models used are proprietary and extremely large, making them less practical for certain deployment scenarios. This raises a critical question: can we retain the generalization capabilities of PPP while reducing computational cost? In the next chapter, we explore this question in the context of relation extraction by investigating how smaller instruction-tuned models can be used effectively within the PPP framework.

6

APPLYING INSTRUCTION-TUNED LARGE LANGUAGE MODELS FOR RELATION EXTRACTION

In this chapter, we explore whether it is possible to retain the generalizability of large language models (LLMs) while tuning them for a specific task, relation extraction. While the previous chapter has demonstrated the promise of the Pretrain, Prompt, and Predict (PPP) paradigm, the reliance on proprietary, large-scale models raises questions about accessibility, control, and deployment cost.

To this end, we investigate the effectiveness of instruction tuning for a smaller, open-sourced 7B model, asking:

RQ4 *To what extent can we improve Pretrain, Prompt, and Predict-based models to perform Relation Extraction?*

By tuning the model using parametric-efficient methods on natural language instructions and relation extraction examples, we aim to assess whether instruction-tuned LLMs can match the performance of larger models while remaining lightweight and adaptable under the PPP paradigm.

6.1 INTRODUCTION

Large language models (LLMs) have exhibited impressive performance across various NLP tasks. Using the in-context learning (ICL) paradigm, wherein models are shown demonstrations to handle new tasks without updating any model parameters, LLMs have showcased performance on par with fully supervised smaller language models (LMs)[1] (such as BERT-based models) while using a limited number of examples [129].

Despite this notable achievement, previous studies have shown that LLMs using ICL still significantly underperform when compared to fully supervised smaller LMs, particularly for relation extraction (RE) [91, 206, 233]. RE represents a fundamental and challenging building block within Information Extraction (IE) pipelines, as it requires semantic understanding of sentences to extract subject-predicate-object triples, which are essential for knowledge base construction (KGC). This limited performance might stem from the low incidence of RE tasks in the dataset used to train the LLMs [233].

To overcome performance deficits when using ICL with LLMs, instruction-tuning [146] can present a different approach to harness the capabilities of

1 We refer to smaller LMs as LMs that have under 1B parameters

LLMs. This involves fine-tuning LLMs on datasets where tasks are described using instructions.

However, as the number of parameters grows, updating these model parameters while working with resource constraints becomes increasingly impractical. To address this, the development of parameter-efficient fine-tuning (PEFT) techniques allows models to adapt to different domains or tasks without consuming excessive time or space. PEFT methods typically modify only a smaller number of additional tunable parameters while keeping the primary model parameters frozen. Two widely adopted PEFT approaches are prefix-tuning [118] and low-rank adaptation (LoRA) [83].

To this end, we explore how instruction tuning can help improve LLMs on RE tasks, using the LoRA technique. Importantly, the injectable learned lower-rank matrices allow us to efficiently adapt to a new task or domain both in time and space. Besides being efficient, this also allows for the retention of properties that are frequently found useful in instruction-tuned models, e.g. the ability to respond to chat-style conversational input or to answer factoid questions.

Specifically, we fine-tune an open source LLM called *Dolly-v2-3B* [42], using the LoRA approach, on a silver standard RE dataset [86] that has been transformed into an instruction-based dataset. The dataset contains around 1,079 different types of relations, making it challenging for smaller-sized LMs.

We applied two evaluation approaches. First, we evaluated the model using an exact match with silver standard labels. Results show the model achieves a 28.5% micro-F1 score and a 27.3 macro-F1 score. After qualitative investigation, we observed that the model generates a substantial number of correct triples that are not included in the dataset annotations. To better assess the model's true performance, we randomly sampled 100 instances from the test set and manually evaluate the triples produced by our model. Furthermore, we also observed that some generations contain correct triples but cannot be derived from the input text (we refer to them as out-of-scope triples). Our hypothesis is that Dolly-v2-3B was fine-tuned on Wikipedia-related data; hence, the model contains related knowledge. Therefore, we apply two criteria for our manual evaluation: (1) if the triple is correct; (2) if the triple can be derived from the input. Our results show an average accuracy of 66.5% with a 0.742 Cohen's Kappa inter-agreement. In addition, the results also indicate that 8.5 % of triples are out-of-scope triples.

In summary, our contributions are threefold:

- Code to transform existing RE datasets into instruction-based datasets [2].
- An instruction-tuned Dolly-v2-3B model capable of performing relation extraction.
- An evaluation of this model performance using both an existing relation extraction baseline dataset complimented with a manual analysis. These initial

2 https://github.com/INDElab/KGC-LLM.git

results indicate that instructed models can potentially be competitive with fully supervised models using less annotated data.

6.2 RELATED WORK

Relation Extraction: Over the years, several different approaches have been framed for RE. Early approaches treated RE as a pipeline involving named entity recognition followed by relation classification [229]. More recently, end-to-end approaches leveraging the transformer architectures have been used [214]. Additionally, attempts to employ seq2seq models for RE have gained attention and led to significant improvements [86, 94].

A key challenge is to extract large numbers of entities and relations. For example, the REBEL dataset [86] contains over a thousand types of relations. To tackle this scale, generative models have been employed. A state-of-the-art example is GenIE [94] which frames RE as a Generative Information Extraction task and employs a constrained decoding strategy. Training on the REBEL dataset, GenIE achieved a 68.93 micro-F1 score and 30.46 macro-F1 score.

LLMs with ICL for RE: Despite the high performance of LLMs on various tasks, previous work attempted to explore their performance on RE using ICL [91, 206, 233]. The results indicate that LLMs are not good few-shot learners when it comes to RE. For instance, Jimenez Gutierrez et al. showed that LLMs underperform smaller LMs for biomedical RE.

Instruction-tuning: Recently, supervised fine-tuning on a large number of tasks represented with demonstrations has shown improvements in LLMs' capacity to generalise to unseen tasks [213]. To better exploit knowledge learned by LLMs during pre-training, different adaptation strategies have been developed to make fine-tuning LLMs more practical. For example, prefix-tuning [118] updates only a small part that is the prefix of pre-trained transformers while keeping the rest of the model parameters frozen. LoRA [83] proposes a low-rank adaptation fine-tuning strategy that does not modify the model itself but instead trains injectable lower-rank matrices. An additional advantage of LoRA is that it can used with other strategies, such as prefix-tuning.

Our work differs from the aforementioned in that we transform classic RE datasets to instruction-based datasets and then instruction-tune an LLM using LoRA. Importantly, our data transformation strategy allows any RE dataset to be transformed and fine-tuned with any LLM.

6.3 METHODS

Data Transformation: We convert the REBEL dataset to an instruction-based dataset for fine-tuning LLMs. Unlike building an instruction-based dataset with different tasks[212, 213], our transformed dataset focuses solely on one task: ex-

Instruction: A triple has three components: (subject, relations, object).
Extract triples from the given text.
Input:
Eri-TV is a state-owned Eritrean television station.
Output:
Extracted triples are: (['Eri-TV', 'country', 'Eritrea'],)

Figure 6.1: One example of a demonstration of the transformed RE dataset.

tracting triples. The prompt template we utilize is adapted from [212], which comprises three components: *Instruction*, *Input* and *Output*. An example instruction can be seen in Figure 6.1. The REBEL training set consists of 3,120,296 samples and 1,079 different types of relations, which we convert.

Instruction-tuning: We proceed to instruction-tune a Dolly-v2-3B model with LoRA. Dolly-v2 [42] is a series of open-sourced large language models based on Pythia-12b, which were instruction-tuned on 15k instructions generated by employees of Databricks Dolly-v2 models are available in different parameter sizes, ranging from 3B to 12B. Considering computational constraints, we select the 3B model for our experiments. Dolly-v2 has been instruction-tuned on Wikipedia data with questions that required contributors to refer to specific information from given Wikipedia paragraphs. This attribute makes the model well-suited for our RE dataset, given that the texts in the REBEL dataset are also sourced from Wikipedia.

Evaluation: We evaluate the performance of the instruction-tuned model on RE using two distinct methods:

- **Traditional evaluation.** For the traditional evaluation, we strictly match the subject, relation and object triple with gold labels. Subsequently, we calculate the precision, recall and F1 score, both micro and macro, under the assumption that the labels in the REBEL dataset are fully complete and correct.
- **Post-hoc human evaluation.** For the post-hoc human evaluation, we follow the evaluation methods presented by Groth et al. and Wadhwa, Amir, and Wallace, where human annotators judge the output of model Each triple is assessed based on two criteria: (1) whether the triple is correct or not; (2) whether the triple is correct but cannot be inferred from the provided sentence. The first criterion assesses the precision of the model's generation, while the second one gauges the model's ability to generate correct triples from its background knowledge. We term such correct triples "out-of-scope" triples.

6.4 EXPERIMENTS AND RESULTS

In our experiments, we employed specific hyperparameters, namely the number of epochs and the ranks of the matrices in LoRA. Ultimately, we conducted

our experiments with 3 epochs and a rank of 4, which aligns with the numbers used by Hu et al. To determine the best-performing model, we evaluated the models on a validation set containing 50 samples due to inference time constraints. Subsequently, we selected the best-performing model to evaluate and report final performance results.

The results of the strict evaluation can be found in Table 6.1. Notably, the state-of-the-art model outperforms the instruction-tuned model under the assumption that the provided annotations are complete and correct. However, when we assess the precision as evaluated by humans (as shown in Table 6.2), we observe that the precision is around 66.5%. In both human evaluation criteria, the inter-agreement between the two evaluators exceeds 0.7, indicating a substantial level of agreement between evaluators. We also note that 8.5% of the triples in the human evaluation what we term out-of-scope, namely, they were correct but not entailed by the given sentence.

	Micro			Macro			# of instances
	Precision	Recall	F1	Precision	Recall	F1	for training
GenIE	68.02	69.87	68.93	33.9	30.48	30.46	3,120,296
Instruct-tuned Dolly-v2-3b	36.6	23.3	28.5	36.7	22.6	27.3	102,400

Table 6.1: Results of strict evaluation of instruction-tuned model vs. the state-of-the-art.

	Value	Cohen's kappa
Precision	66.5	0.760
Out-of-scope rate	8.5	0.724

Table 6.2: Results for human evaluation with two evaluators on randomly sampled 100 instances from the test set.

6.5 DISCUSSION

Training Data Amount: It is important to note that the reported model is based on 800 steps of fine-tuning, equivalent to 102,400 samples, making up only 33% of the training set, while GenIE is trained on the full dataset. Using a higher rank of adaptation model might be able to improve the performance further.

Disparity between Micro and Macro Measurements: An intriguing observation is the significant performance disparity between micro and macro measurements for GenIE. This indicates that the model performs poorly on certain types of relations but better on others, which could be attributed to the existence of long-tail relations. It is likely that GenIE performs well on dominant and frequent relation types but not so well on less frequent relation types. In contrast,

the instruct-tuned model exhibits consistent performance between micro and macro measurements suggesting that it struggles less with long-tail relations.

Performance Increase for Human Evaluation: During the analysis, we noticed that many triples generated by the instruction-tuned model are correct but not included in the dataset annotations. Out of 100 random samples, the instruction-tuned model generates 453 triples. Among these 453 triples, both evaluators agree that 274 triples are correct. Interestingly, when comparing these triples with the dataset annotations, we found that 184 of the human-evaluated correct triples are not included in the REBEL annotations, accounting for 67.2% of the correct triples generated by the model. This finding indicates that the current evaluation approach might have limitations when applied to instruction-tuned models. Evaluating generative LLMs remains a challenge, also for tasks such as mention detection [50]. Moreover, it is noteworthy that the model demonstrates the ability to generate out-of-scope triples, indicating that its generation process relies on both the input context and the knowledge learned from pre-training.

6.6 CONCLUSION

Our findings demonstrate the potential of instruct-tuned models for RE, especially when dealing with a substantial number of relations. Even with a 3B model (considerably smaller than the 176B parameters of GPT-3), fine-tuned on a relatively small amount of data, the model already exhibits good performance for RE. We anticipate that further exploration and fine-tuning will likely lead to even better performance. Furthermore, the instruction-tuned model displays the ability to generate out-of-scope triples to a certain extent, indicating that the model retains knowledge acquired during its pre-training, which holds promise for unifying LLMs and knowledge graphs. Finally, our methods are generalizable and can be applied to any existing RE datasets, underscoring their applicability and potential for future research.

Having shown that instruction-tuning a relatively smaller model can effectively teach it to perform specific tasks, we now turn our attention to orchestrating different LLMs to strike a balance between accuracy and cost. In particular, we explore strategies for selectively leveraging very large LLMs only when necessary while relying on code generation for the bulk of the cases. The next chapter investigates how such orchestration can enable scalable and cost-effective systems for data preparation.

7

EFFICIENT DATA WRANGLING WITH LARGE LANGUAGE MODELS USING CODE GENERATION

Beyond extraction, data preparation is a critical stage of the Knowledge Graph Construction (KGC) pipeline. Integrating heterogeneous sources often requires entity matching, cleaning, and transformation to ensure and maintain the quality and completeness of the constructed knowledge bases [81, 218]. These tasks are a key phase in the KGC lifecycle. Pipelines often move from extraction to integration/fusion and then finally completion. Quality checks often lead to iterative cleaning and de-duplication before deployment.

Therefore, in this chapter, we continue our investigation of PPP-based solutions for data preparation tasks, particularly data wrangling, by designing a system that balances cost and accuracy. We formulate the following research question:

RQ6 *How can PPP workflows improve cost-efficiency for data preparation tasks?*

To address this, we propose a hybrid framework that leverages large language models for strategic decision-making and code generation, while minimizing redundant usage of expensive models. Our system is applied to a range of data wrangling tasks, including error detection, entity matching, value imputation, and data transformation, demonstrating both effectiveness and cost-efficiency across benchmarks.

7.1 INTRODUCTION

Efficient and accurate data wrangling is fundamental to modern data-driven applications [136, 156]. Studies consistently report that data scientists spend the majority of their time on cleaning, transformations, and entity matching tasks, such as re-formatting timestamps, normalizing inconsistent entries, correcting errors, and integrating heterogeneous data sources [77, 79, 156]. As datasets scale in size and heterogeneity, this “data plumbing" becomes a dominant bottleneck, diverting efforts from implementing higher-value tasks such as modeling and analysis.

While data wrangling has been addressed through a variety of approaches, two lines of work particularly relevant to our study are programming by example (PBE) and applying large language models (LLMs) on a per-row basis. **PBE systems** [23, 78] synthesis transformation programs given user-provided input-output example pairs. PBE excels in syntactic tasks by overfitting to the provided I/O pairs, but its domain-specific languages and reliance on struc-

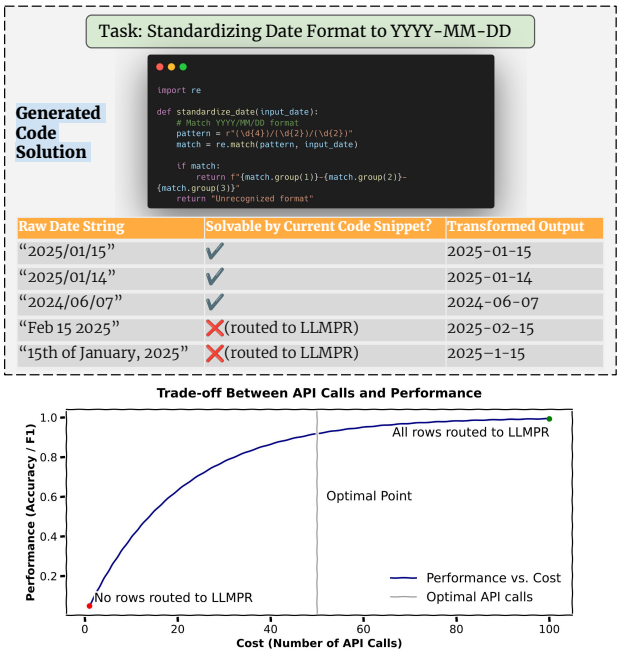

Figure 7.1: The example above shows when a row can use a given generated code solution and when LLMPR is most appropriate. The plot below illustrates the conceptual trade-off between API calls and performance in TableSwift introduced by the data router. The curve illustrates the diminishing returns of increasing API calls, while the optimal point marks the ideal balance between cost and performance.

tured examples limit its coverage, especially in cases when semantic reasoning or domain knowledge is required.

On the other hand, LLMs excel at reasoning about complex and semantically rich data wrangling tasks by directly interpreting natural language instructions [30]. When applied on a per-row basis (referred to as **LLMPR** in subsequent sections), LLMs can achieve state-of-the-art accuracy [151]. However, prompting each record (or per pair in entity matching) results in cost and latency growing at least linearly with data volume. Moreover, row-wise prompting is non-deterministic and opaque, complicating debugging and reproducibility. Hence PBE is fast but brittle, whereas LLMPR is accurate but expensive, highlighting a cost-accuracy trade-off. One simplified example is shown in Figure 7.1,

Recent work aims to optimize LLM-based data wrangling workflows [34, 119, 130] by strategically leveraging the semantic power of LLMs. For example, AutoDW [130] utilizes LLMs to recommend the target variable, ML tasks, and to generate plans to fill parameterized code templates. However, the predefined templates still need a substantial amount of manual effort. Furthermore, the system does not perform instance-level validation, resulting in a single plan being applied uniformly across all rows, regardless of edge cases. SEED [34]

utilizes LLMs to generate execution plans that combine code generation, vector caching, and small models, and further optimizes the plan based on cost estimation over a held-out set. However, once the plan is generated, no more code candidates are generated, even if a large portion of residual rows remains unhandled, especially when data has a high degree of variability.

Real-world tables are often messy and contain diverse logic and surface forms that require more than one generated code snippet. The ability to dynamically generate multiple code solutions is thus crucial for covering such heterogeneous patterns.

To bridge this gap, we propose **TableSwift**, a hybrid framework that integrates LLM-based code generation with LLMPR through an explicit lightweight data routing component. While the use of LLMs for code generation tasks, such as text-to-SQL, has been extensively studied [19, 115, 236], generating executable, high-quality code in a single attempt remains challenging. This difficulty arises due to the complexity and the variability of the input samples, as well as the sensitivity of code correctness to small generation errors [216]. Therefore, in order to ensure the code quality, TableSwift employs a generate-verify-rank (GVR) pipeline to synthesize reusable Python or DuckDB SQL programs from natural language instructions. By generating multiple candidate programs and validating them against demonstration data, GVR provides refinement signals (runtime feedback) that guide the selection of higher-quality code.

TableSwift employs an explicit data router that determines whether each instance satisfies the input assumptions of the current code solution. Instances not covered by the current solution are routed for further processing, enabling the system to iteratively generate and verify additional code candidates to maximize coverage. To minimize manual efforts, the data router is generated dynamically alongside each code solution, aligning the decision logic directly with the input constraints embedded in the code. Looking at the example in Figure 7.1, the current code solution can resolve the majority of the inputs, while those that do not fulfill the input requirement are routed to use LLMPR. This design navigates the trade-off shown in the plot below in Figure 7.1. The conceptual curve between cost and performance relies on the two assumptions. One is that LLMPR is able to achieve high performance on various data wrangling tasks [151]. The other is that some of the rows can be handled by code effectively [119].

The best-case scenario occurs when the generated code processes all rows, achieving the highest performance with minimal API calls. Conversely, the worst-case scenario occurs when the data router routes all rows to LLMPR, resulting in high costs and low accuracy. Even in the worst-case scenario, TableSwift maintains the same cost as LLMPR when scaled to large datasets without introducing any extra overhead. Between these extremes lies a spectrum where code solves some portion of the dataset, with semantically complex rows

being routed to LLMPR. As the fraction of rows routed to LLMPR increases, costs rise due to the additional API calls, while accuracy may initially improve as semantic reasoning addresses harder cases. However, this relationship is non-linear: when generated code handles the majority of simpler rows, the trade-off results in a desirable balance between performance and cost. This paper explores these scenarios across tasks, identifying when TableSwift achieves its best-case efficiency and when it falls back to LLMPR dominates due to task complexity.

Furthermore, we introduce a formalized cost-efficiency analysis that models the trade-offs between performance and resource usage using a weighted cost function. This quantitative framework provides insights into the benefits of our hybrid approach compared to traditional methods, demonstrating significant cost savings and adaptability across a range of data wrangling tasks.

In summary, the work makes the following key contributions:

- A unified workflow that integrates code generation and semantic reasoning, supported by a lightweight data router, for efficient and adaptable data wrangling.
- A formal cost-efficiency analysis to model the trade-offs between performance and resource usage, providing a comprehensive evaluation of the framework.
- A demonstration of the effectiveness of the proposed approach through extensive experiments across diverse tasks, achieving state-of-the-art performance while reducing costs and ensuring reproducibility.

7.2 RELATED WORK

LLMs for Data Management. LLMs have significantly impacted data management, enabling LLM-powered databases [195], hybrid querying systems [201, 208], and automating data integration and cleaning [127, 149, 151]. Many recent approaches integrate LLM operators into database pipelines, allowing structured queries over unstructured data [4, 128, 159, 185].

However, these methods apply LLMs at query time, leading to high computational costs. Some mitigate this via code generation [4, 128], but lack structured verification, making them error-prone. Our work improves on these approaches by integrating code synthesis with a verification mechanism, ensuring both efficiency and correctness.

Data Wrangling with LLMs. LLMs have demonstrated strong capabilities for data transformation, error detection, and imputation. Row-wise inference [89, 151] allows fine-grained, context-aware transformations, but suffers from high latency, non-determinism, and cost. For large datasets, the number of API calls scales poorly, making this approach impractical.

Several approaches attempt to mitigate these costs. SEED [34] combines LLM prompting, small model fine-tuning, and caching to optimize execution plans. However the code generation terminates once the plan is generated and optimized, falling short when there are high variability in the input data. AutoDW [130] utilizes LLMs to determine target variables and identify whether the task is classification or regression, generating plans for existing code templates that still rely on predefined parameterized templates.

Another direction uses parameter-efficient fine-tuning [202, 223, 232, 235], showing that small models can outperform large LLMs for entity matching. However, these methods rely on static pipelines and struggle with diverse transformation tasks.

Our approach reduces per-row computation while maintaining flexibility in handling various tasks and scalability by dynamically generating multiple solutions.

Program Synthesis in Data Management Transform-Data-by-Example (TDE) frameworks [23, 31, 78] generate transformation programs from user-provided examples, making them effective for structured, rule-based changes like reformatting dates. However, they struggle with semantically complex tasks and lack support for natural language instructions, limiting their applicability.

LLM-based code generation methods [30, 40] improve flexibility by synthesizing transformation programs from natural language descriptions. Despite their potential, these approaches often lack verification mechanisms, requiring users to manually validate results. Small inaccuracies in generated SQL queries or Python functions can lead to significant errors across datasets.

Recent work has explored improving LLM-generated transformations. Huang et al. [84] enhance accuracy by integrating textual, code, and data context in computational notebooks. Barke et al. [8] introduce SOFSET, a dataset of real-world NL-to-code tasks from StackOverflow, proposing a cluster-then-select prompting technique to improve the generalizability and correctness of LLM-generated Pandas code. These findings underscore the sensitivity of LLM performance to input structure.

7.3 PROPOSED APPROACH: TABLESWIFT

In this section, we introduce TableSwift, our hybrid data wrangling system. To address the limitations of row-wise execution using LLMs, TableSwift combines code generation with a fallback LLMPR solution via a data router. The key idea behind this framework is to generate task-specific functions that can handle a majority of the data rows while routing unsolvable instances to LLMPR for direct processing. This shift allows us to leverage LLMs strategically, utilizing their strong semantic reasoning capabilities only where necessary. The code generation component ensures transparency by producing context-aware, vali-

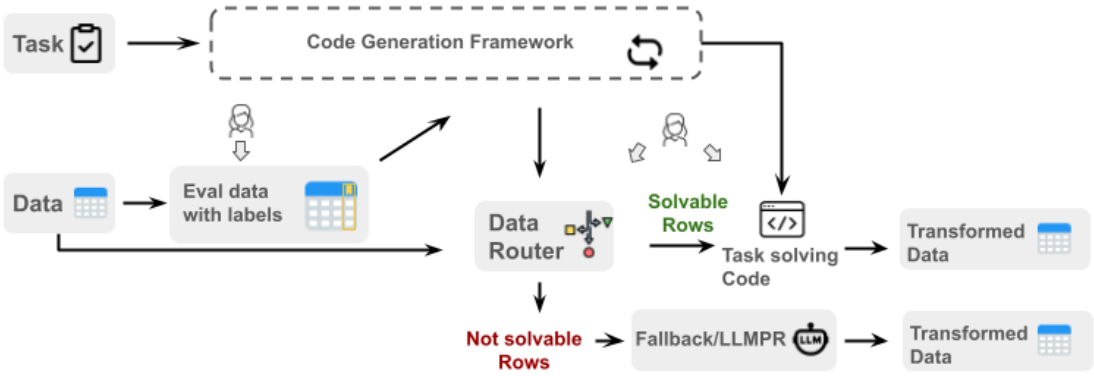

Figure 7.2: Overview of TableSwift.

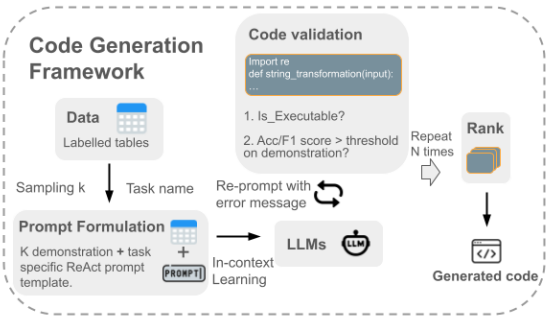

Figure 7.3: Code generation framework.

dated functions that are adaptable to new languages and domains. Meanwhile, the data router component provides a fine-grained dynamic optimization at the row level. Unlike static rule-based routing, the data router is conditioned on the current code solution and generated on the fly.

7.3.1 *High-level overview of the approach.*

As shown in Figure 7.2, a task instruction, data, and evaluation data containing labeled examples are provided to the system. The evaluation data is typically small on the order of 3-10 examples. This evaluation data helps the system capture the intended logic while accounting for common patterns in data, and serve as signals for evaluating generated code. During application, the task instruction and labeled examples can be provided by the user. Then, the system generates code snippets tailored to the task and dataset using the code generation framework. Each resulting code snippet has been validated to be executable and evaluated on the provided labeled examples. Since generating high-quality code is essential for minimizing the reliance on LLMPR, this pro-

cess is iterated multiple times to produce the highest-quality code. Once a suitable code solution is obtained, an instance of the data router is dynamically generated, conditioned on the current code solution. The task-specific router filters rows that are not valid as inputs for the code snippet. If a significant number of unsolvable rows remain, the system iterates, either by re-sampling from the training set, or expanding the example coverage by the user. Until we reach the pre-defined maximum number of solutions or only a handful of rows left to process by LLMPR. Finally, we aggregate the transformed outputs from both applying code-based solutions and LLMPR, integrating deterministic transformations with semantic reasoning where needed.

7.3.2 *Code Generation Framework*

The code generation framework in Figure 7.3 is a key component of TableSwift. It leverages LLMs to generate reusable task-specific code. The framework is designed to be language-agnostic, allowing for adapting to multiple programming languages. In this work, we focus on Python for its popularity and flexibility and DuckDB SQL for its efficiency in structured data processing.

The code generation framework employs the **GVR (*Generate, Verify, Rank*)** procedure. This systematic approach ensures the generated code is not only functionally correct but also optimized for task-specific performance. The GVR logic is generalizable to applications when programming by examples with LLMs.

Generate. The *generate* step involves prompting LLMs to produce task-specific code. This step can guide the models to create code solutions tailored to the task, dataset, and user requirements, as well as refine the code based on system feedback.

PROMPT FORMULATION The prompts consist of three main components. The general prompt features, the task-specific instruction, and the demonstration. The prompt adopts the ReAct framework [226], instructing models to perform reasoning before generation. This can reduce hallucination throughout the generation process.

The general prompt features provide an overall instructional message to the model, guiding it to reason about the task first. We also specify language-specific instructions as one of the features. For the task-specific instruction, we provide any task-related information such as detailed task descriptions (e.g., convert Celsius to Fahrenheit.), preferred methods (e.g., regex, similarity measure), and any specific considerations (e.g., avoiding training machine learning classifiers). One example prompt is shown in Figure 7.4.

The task-specific variants are as follows for the tasks considered in this paper:

- Data Transformation (DT): uses regex and string manipulation to transform input-output pairs, encouraging handling edge cases.
- Entity Matching (EM): incorporates similarity functions and defining dynamic threshold, striving to create generalizable functions to avoid overfitting.
- Data Imputation (DI): uses logical conditions and dynamic ranges, ensuring generalization across the dataset.
- Error Detection for spelling (DT): focuses on detecting spelling errors while avoiding exact string matches, suggesting incorporating domain-specific vocabularies.

Lastly, we provide labeled input-output pairs as demonstration data. In this work, we use random sampling as our strategy for sample curation for the relatively small-scaled benchmark datasets.

The advantage of the current prompt design is that the structure is language-agnostic, making it adaptable to different programming languages. Moreover, the specific structure enables iterative improvements, ensuring high-quality code generation. Last but not least, the task-specific variant allows for optimizing and customizing the instructions and demonstrations, ensuring the flexibility to adapt to different tasks.

Verify. The *verify* step provides guarantees for the generated code to be functionally correct and meet task-specific performance expectations. Therefore, we apply a 2-dimensional validation process: syntactic correctness and functional accuracy. We first verify that the generated code can execute without errors on a given input, catching issues such as syntax errors, undefined variables, type mismatches, and potential runtime failures. Additionally, to prevent infinite loops or excessive computation, we enforce an automatic destruction mechanism that involves a timeout execution strategy, terminating any code that fails to complete execution within a reasonable timeframe. If the execution fails, we capture error messages from the compiler as error messages. Additionally, we employ self-defined error messages such as timeout violations. Then, we assess the functional accuracy of the code by evaluating the performance on the provided demonstration examples. The performance is calculated against the ground-truth labels, computing task-dependent metrics such as accuracy or F1 score. A performance threshold, set as a hyperparameter, determines whether the generated code is sufficiently reliable for deployment. If the performance falls below the threshold, we employ a self-defined error message template and subsequently use it as feedback.

When failures occur at either stage, TableSwift compiles the relevant error messages, including syntax errors, runtime exceptions, timeouts, or low performance, into structured feedback. This feedback is then fed back into the LLM to initiate self-refinement [135], guiding the model to produce an optimized solution.

Rank. Generating a single function may not always yield the best solution, as LLM-generated outputs has inherent variability depending on the provided demonstrations [8] and the stochastic nature of sampling. To improve robustness, TableSwift does not rely on a single generated function; instead, it iterates through multiple *generate* and *verify* cycles to create a pool of candidate functions. Once a diverse set of validated functions is obtained, the *rank* step selects the best-performing function. The primary criterion for ranking is the accuracy/F1 score associated with each code solution. Additionally, in cases where multiple functions achieve similar accuracy, we can also evaluate execution efficiency and function understandability by either human experts or pre-defined constraints. The ranking mechanism introduces an implicit optimization process, ensuring the final selected function is not only valid but also preferable given user-defined constraints. TableSwift increases the likelihood of selecting a function that generalizes well across different data distributions.

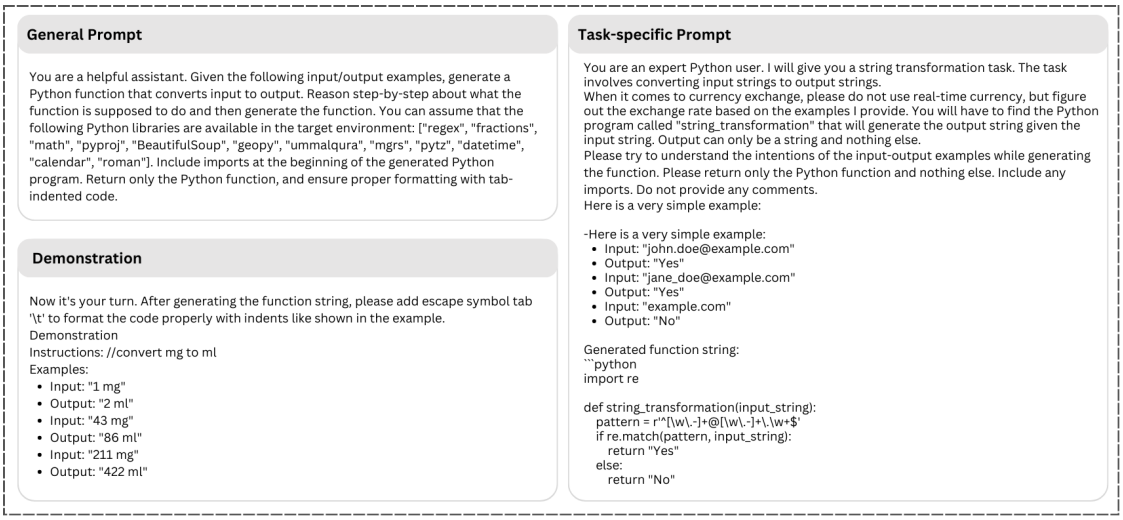

Figure 7.4: One example prompt consists of the general system prompt, task-specific instruction, and labeled demonstrations. The instruction describes a string transformation task - convert mg to ml. LLM is prompted to generate Python code.

7.3.3 *The Data Router Component*

The data router is the core of TableSwift's efficiency and adaptability, which dynamically determines whether a row can be processed by the generated code, aiding in identifying edge cases, irregularities, and semantically complex inputs. The data router provides routing decisions to balance cost and accuracy, enabling TableSwift to handle heterogeneous data distributions robustly.

Motivation A naive approach would directly apply the generated transformation code by Text-to-SQL method to all rows, but this leads to a significant

Algorithm 1 Generate Router for Current Code Solution

Require: Verified code solution f_{code}, calibration set $D_{val} = \{x_i\}_{i=1}^{N}$
Ensure: Router function $R(x) \in \{\text{ACCEPT}, \text{REJECT}\}$
  for all $x_i \in D_{val}$ **do**
    Execute $f_{code}(x_i)$
    if $f_{code}(x_i)$ executes successfully **then**
      $y_i \leftarrow 1$
    else
      $y_i \leftarrow 0$
    end if
  end for
  $L \leftarrow \{(x_i, y_i)\}$ ▷ labeled demonstration data
  Use LLM to synthesize candidate router rule $R(x)$ predicting y_i from x_i
  return $R(x)$

problem: applying the same logic across a heterogeneous dataset disregards data variability. The key challenge is that datasets often contain subgroups of records that require different data wrangling logic, and a one-size-fits-all code solution is often insufficient. Therefore, the data router builds the guardrails, validating the suitability of the row before applying the code.

Iterative Code Generation with Routing. TableSwift operates through an iterative process that alternates between code generation, verification, router synthesis, and residual reduction. At iteration t, given the residual dataset $D_{res}^{t} \subseteq D$, the system generates a set of M candidate transformation programs $\{f_t^{(j)}\}_{j=1}^{M}$ using an LLM conditioned on the task instruction. Each candidate is evaluated on a labeled demonstration subset $D_{dem}^{t} \subset D_{res}^{t}$, and the candidate with highest verified accuracy is selected as f_t.

Next, a router R_t is synthesized for f_t using execution traces on D_{dem}^{t}, as detailed in Algorithm 1. The router R_t identifies which instances satisfy the preconditions of f_t, allowing safe application of the generated code. The residual set for the next iteration is defined as:

$$D_{res}^{t+1} = \{x \in D_{res}^{t} \mid R_t(x) = \text{REJECT}\}.$$

The iterative process terminates when $|D_{res}^{t}| \leqslant \epsilon \cdot |D|$, where ϵ is a user-defined stopping threshold controlling the acceptable fraction of unresolved rows.

Implementation Strategies The data router can be implemented in multiple ways. However, the inherent requirement is to be cost-efficient and task-aware. Therefore, in TableSwift, the data router is dynamically generated alongside the wrangling code rather than being pre-defined. This makes it adaptable to different datasets, task formats, and, most importantly, the generated wrangling code of interest. Instead of relying on classifiers that require domain-specific

tuning and introduce extra overhead, we leverage LLMs to generate router functions. These functions typically take the form of regular expressions(regex) or structured validation functions, which can quickly determine whether an input adheres to the assumptions of the code solution. For instance, in a date format conversion task like the one in Figure 7.1, the router function may use regex validation to check whether an input follows the expected *"YYYY/MM/DD"* pattern before applying a code-based transformation to *"YYYY-MM-DD"*. If an input does not match the pattern, perhaps due to ambiguous formats, missing values, or incorrect delimiters, it is routed to not use the current generated code solution. Then, we can either come up with another code solution when the rows left are significant or route to LLMPR eventually. An additional strength of TableSwift's router design is its ability to iteratively solving a subset of the dataset rather than relying on a single wrangling function. For example, after the first round of processing, the system can generate a second function to handle the remaining subset of rows. This process continues until the rest of the data exceeds the user defined fraction threshold.

7.4 COST-EFFICIENCY ANALYSIS

The cost-efficiency of TableSwift is evaluated based on the number of API calls required to process a dataset of size N. We compare the hybrid approach against the LLMPR-only method, considering best-case, worst-case, and intermediate scenarios. A key distinction arises in Entity Matching (EM) tasks, where the cost of LLMPR is quadratic due to pairwise comparisons.

7.4.1 *LLMPR Cost Model*

In an LLMPR-only approach, the total number of API calls depends on the task:

- **For row-wise tasks (DT, DI, ED):** Each row is processed independently, leading to a **linear** cost:

$$\text{API Calls (LLMPR, Row-wise)} = N$$

- **For Entity Matching (EM):** Each row must be compared against multiple other rows, resulting in a **quadratic** cost:

$$\text{API Calls (LLMPR, EM)} = O(N^2)$$

This makes EM particularly prohibitively expensive at scale, as API calls grow exponentially with dataset size.

7.4.2 *TableSwift Cost Model*

TableSwift mitigates LLMPR costs by leveraging code generation to process the majority of rows and selectively routing only difficult cases to LLMPR. The number of API calls depends on the proportion of rows routed, as well as the number of trials (T) and retries (R) required for code verification.

BEST CASE (UPPER BOUND) In the best-case scenario, all rows are successfully processed by the generated code, and no rows are routed to LLMPR. The total number of API calls is dictated solely by code generation trials:

$$\text{API Calls (Best Case)} = T \times R$$

Since this cost is independent of N, TableSwift exhibits constant cost scaling, which is particularly advantageous for large datasets.

WORST CASE (UPPER BOUND) In the worst-case scenario, all rows are routed to LLMPR, reducing TableSwift to an LLMPR-only system:

$$\text{API Calls (Worst Case)} = \begin{cases} N & \text{for row-wise tasks (DT, DI, ED)} \\ O(N^2) & \text{for EM (pairwise comparisons)} \end{cases}$$

This confirms that even in the worst case, TableSwift incurs no additional cost compared to LLMPR-only execution.

INTERMEDIATE CASE (PRACTICAL SCENARIO) In the practical case, a fraction f of rows is routed to LLMPR, while the remaining $(1 - f)$ rows are processed by the generated code. The total number of API calls is:

$$\text{API Calls (Intermediate)} = fN + (1 - f) \cdot (T \times R)$$

Since f varies based on dataset complexity, this scenario represents TableSwift's adaptive trade-off between efficiency and accuracy.

7.4.3 *Weighted Cost-Efficiency Analysis*

To quantify TableSwift's balance between cost and performance, we define a weighted cost function:

$$\text{Cost} = \alpha \cdot \frac{1}{P} + \beta \cdot C$$

where:

- P is the task performance (e.g., accuracy or F1 score).
- C is the total API call count.

- α, β control the relative emphasis on performance vs. efficiency.

This formalization captures TableSwift's advantage: minimizing API calls while maintaining strong task performance. The data router plays a pivotal role, dynamically optimizing the system's cost-performance balance.

7.5 EXPERIMENTAL SETUP

In this section, we describe the setup for evaluating the hybrid framework across various tasks, models, and configurations. The objectives of these experiments are four-fold:

- Assessing performance of the code generation approach using both Python code and DuckDB SQL code on Data Transformation.
- Investigating the performance of the data router component.
- Experimenting to what extent the hybrid framework can address other Data Wrangling tasks.
- Analysing the trade-off in cost and accuracy.

Tasks Evaluated. We mainly focused on Data Transformation tasks while additionally, we tested the framework on three other data-wrangling tasks: Error Detection (ED), Data Imputation (DI), and Entity Matching (EM). The definitions of these four tasks are as follows.

- Data Transformation (DT): Transforming raw input data into a desired format or structure, often involving logic-based or syntactical operations, while sometimes require external domain knowledge.
- Error Detection (ED): Identifying erroneous or invalid entries in datasets, including structured and semantically complex errors.
- Data Imputation (DI): Filling in missing or incomplete values in datasets.
- Entity Matching (EM): Identifying semantically equivalent entities across datasets, often requiring reasoning about varying syntax and domain-specific knowledge.

Types of Code. We test the performance of the code generation approach using two types of code: Python and DuckDB SQL. The motivation for choosing Python is due to its popularity, training data exposure, and rich ecosystem. Python is one of the most widely used programming language for data science and analysis. Due to its popularity, LLMs are more likely to be exposed to Python code. The familiarity means that LLMs can generate Python that is more likely to be able to solve complex tasks. Furthermore, the diverse libraries in Python's ecosystem contribute to this capability. DuckDB SQL was chosen for its efficiency in real-world applications. Meanwhile, as a relatively new system, we assume that DuckDB SQL has had less exposure on the web and thus likely to less reflected in an LLM's training data. This limited exposure provides an interesting lens to evaluate the reasoning ability of LLMs in the context of

data transformation. By comparing Python and DuckDB SQL, we aim to evaluate LLM's code generation ability in the context of Data Transformation when prompted for programming languages with different characteristics.

System Components. To better understanding the effectiveness of the data router component, we tested the system under two configurations: without a data router and with a data router. When the data router is not present, we then apply the generated code to all rows in the test set. While on the other setting, the data router routes code unsolvable rows to LLMPR for additional processing.

Model Tested. To compare the performance of the code generation approach across different models, we evaluate it on the following LLMs.

- GPT-4[154]: a general-purpose enterprise LLM with strong reasoning and code synthesis capabilities.
- Llama3.1-405b[68]: A larger version of the Llama series with 405 billion parameters, designed for advanced reasoning and problem-solving tasks.
- Llama3.2-3b[68]: A smaller-scale, resource-efficient model with 3 billion parameters fine-tuned for various tasks, including content generation and natural language understanding to a certain extent.
- Qwen2.5-coder-32b [165]: A specialized model optimized for the code generation task, which has achieved the best performance among open source code generation language models, with a performance on par with GPT-4o. It is suitable for the approach introduced for its outstanding capacity to generate DuckDB SQL code [240] and its resource efficiency (32 billion parameters).

Hyperparameter Settings. In this work, we configure the following hyperparameters to ensure consistency and reproducibility across all experiments. The maximum number of trials for generating valid code solutions is set to $T = 3$, allowing the framework to make up to three attempts to generate accurate and executable code. Additionally, in cases where the generated code fails to meet the validation criteria, the system performs up to $R = 5$ retries to refine the code. During the verification phase, the validation threshold is set at 51%, meaning the generated code must achieve at least 51% accuracy or F1 score on the demonstration examples to be considered valid. These settings are designed to balance computational efficiency with the quality of the generated code, while ensuring robustness across diverse tasks and datasets.

Evaluation Framework. To evaluate the proposed hybrid approach, we employ both performance metrics and cost-efficiency analyses. Performance is measured using task-specific metrics, such as accuracy for Data Transformation (DT) and Data Imputation (DI), and F1 score for Error Detection (ED) and Entity Matching (EM). For cost-efficiency, we standardize comparisons around the number of API calls required to solve a task, reflecting both financial and computational efficiency.

Additionally, we utilize a weighted cost function introduced in the proposed approach to capture the trade-off between performance and API efficiency. The function is parameterized with weights to emphasize performance versus cost, offering a unified view of system efficiency. This enables a consistent evaluation across tasks and configurations, with experiments designed to highlight the trade-offs of different approaches.

7.5.1 *Research Questions and Experiments*

To evaluate TableSwift comprehensively, we address the following research questions with corresponding experiments:

- **RQ1:** How does TableSwift perform across various data wrangling tasks and datasets?
- **EXP1:** We evaluate TableSwift on all supported tasks—Data Transformation (DT), Error Detection (ED), Data Imputation (DI), and Entity Matching (EM), across a total of 16 datasets. This experiment focuses on assessing the overall performance and adaptability of the system.
- **RQ2:** How do different language models and programming languages affect the performance of the code generation framework on DT?
- **EXP2:** We compare the performance of four LLMs of varying sizes and design purposes, testing two programming languages (Python and DuckDB SQL) on five DT datasets. This experiment investigates the role of model choice and programming language in the code generation framework.
- **RQ3:** How does the data router impact the performance of the system on DT?
- **EXP3:** We conduct an ablation study on the data router component, testing its effectiveness in routing rows to the appropriate processing path. This experiment uses GPT-4 and focuses on the five DT datasets.
- **RQ4:** How does the efficiency-accuracy trade-off of TableSwift compare to LLMPR across tasks?
- **EXP4:** We perform a cost analysis by measuring API calls and accuracy across all datasets to visualize the trade-off between cost-efficiency and performance.
- **RQ5:** To what extent can code generation address various data wrangling tasks, and when should we fall back to LLMPR?
- **EXP5:** We conduct task-specific evaluations and analyze failure cases to identify the limitations of code generation and the scenarios where LLMPR becomes essential. This analysis spans all tasks and datasets.

7.6 EXPERIMENTAL RESULTS

7.6.1 *EXP1: TableSwift on Data Wrangling Tasks.*

Table 7.1: Comparison of TableSwift with baselines on data transformation tasks, measured in accuracy (%). "N/A" denotes not applicable or not reported in their original papers. "CG" denotes the code generation framework. "DDBSQL" denotes DuckDB SQL. "TS" denotes TableSwift. # Rows routed is the number of rows sent to the LLM out of the total number of rows.

Methods	BingQL-semantics	BingQL-Unit	Stack-overflow	FF-GR-Trifacta	Head cases	Average
PBE	32.0	96.0	63.0	**91.0**	82.0	72.8
LLMPR	54.0	65.3	N/A	N/A	N/A	59.65
CG-DDBSQL	73.3	93.0	63.2	69.8	66.7	73.2
TS-DDBSQL	72.0	93.0	73.8	64.0	67.0	73.96
# Rows Routed	5/102	2/99	598/710	24/83	26/90	-
CG-Python	**91.6**	95.0	87.4	83.7	74.6	86.46
TS-Python	90.3	**96.0**	**89.1**	83.6	**85.2**	**88.84**
# Rows Routed	2/102	4/99	584/710	16/83	19/90	-

In this experiment, we evaluate the performance of TableSwift across a diverse set of data wrangling tasks: Data Transformation (DT), Error Detection (ED), Data Imputation (DI), and Entity Matching (EM). Table 7.1 and Table 7.2 summarize the accuracy and F1 score results for DT and other tasks, respectively. The results highlight the adaptability and scalability of TableSwift in handling a range of syntactic and semantically complex challenges.

Data Transformation. TableSwift demonstrates robust performance across DT tasks, achieving new state-of-the-art results on most datasets. In particular, for the challenging "Head Cases" dataset, TableSwift's hybrid design leverages both code generation and the data router, achieving significant gains over standalone code generation and LLMPR baselines. However, for datasets like "BingQL-semantics" and "FF-GR-Trifacta," there are slight performance drops. This suggests potential misrouting by the data router, which we further analyze in EXP3. The breakdown in Figures 7.5 and 7.6 reveals that the majority of rows remain unrouted, demonstrating the strength of the code generation component in solving most cases. For the routed rows, LLMPR successfully addresses several challenging cases, such as in the "Head Cases" dataset, but struggles on others like "StackOverflow," highlighting areas for improvement in the fallback mechanism.

Entity Matching. Entity Matching (EM) presents the most semantically challenging tasks, as it often requires reasoning about domain-specific knowledge and diverse syntax. TableSwift's code generation approach underperforms on datasets like "Amazon-Google," where semantic equivalences vary widely. Despite this, the data router improves performance on most datasets by correctly routing complex cases to LLMPR, showing the potential for incorporating additional expert models in future work.

Error Detection. For Error Detection (ED), TableSwift achieves perfect performance on structured datasets like "Adult" (100% accuracy on the income column), demonstrating its effectiveness in handling well-defined error patterns. However, performance on semantically complex datasets like "Hospital" remains limited (23.5% accuracy), even with the data router. This highlights the need for future enhancements in handling more nuanced error patterns.

Data Imputation. In Data Imputation (DI), TableSwift excels on simpler datasets like "Buy," where the code generation component achieves high accuracy. On more complex datasets like "Restaurant," the data router improves performance significantly (by 22.1%), showcasing the hybrid approach's ability to balance cost-efficiency with accuracy.

Summary. Across all tasks, TableSwift demonstrates its ability to strike a balance between performance and cost-efficiency. The hybrid framework significantly reduces API calls while maintaining competitive performance compared to LLMPR as shown in Figure 7.7. This trade-off is particularly evident in datasets with a mix of solvable and semantically challenging rows, where the data router intelligently routes cases to optimize resource use, such as data transformation datasets. The results emphasize the need for further exploration of task-specific optimizations to enhance the framework's adaptability and performance.

7.6.2 *EXP2: Ablation Studies on Models and Programming Languages.*

Table 7.3 compares the performance of Code Generation (Python and DuckDB SQL) using differing LLMs across several data transformation datasets with the PBE and LLMPR baselines. The results demonstrate a few key findings. First, Python code generation with GPT-4 consistently achieves the best performance across most datasets, including the BingQL-semantics (91.6%) and Stack-overflow (87.4%), which were previously very challenging for the baselines. Second, among the language models trained with varying sizes and intentions, bigger models (GPT-4 and Llama3.1-405b) outperform models with a smaller number of parameters almost at all times in both Python and DuckDB-SQL generation. The results indicate bigger models are better at code generation/reasoning. Qwen2.5-coder-32b has significantly fewer parameters compared to Llama3.1-405b, however, providing somewhat on par or even better

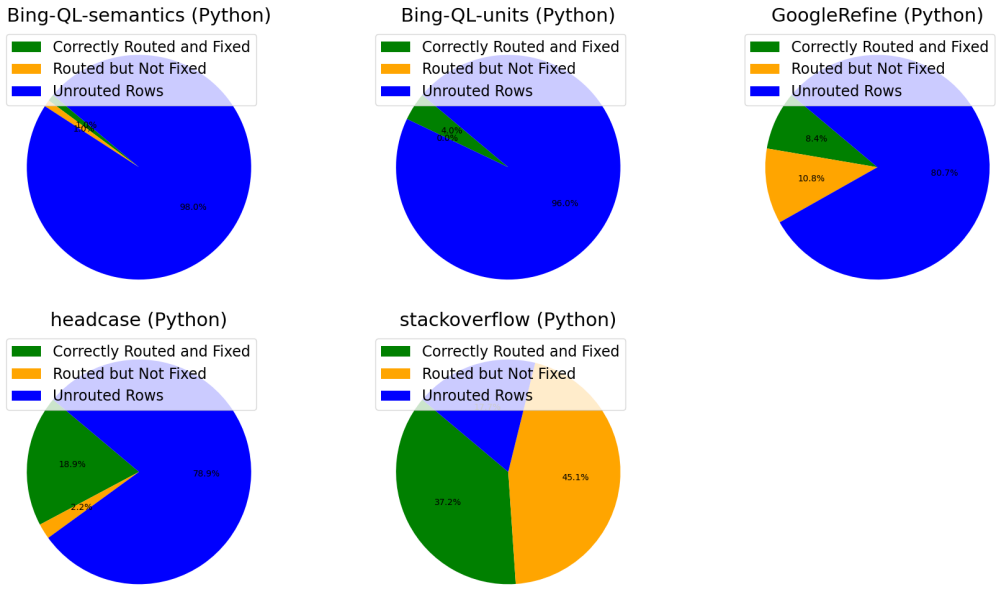

Figure 7.5: Proportions of unrouted rows, routed rows, and correctly transformed rows, in Python code generation.

performance, especially for DuckDB-SQL generation. This result mainly benefits from the model's expertise in code generation. Thirdly, across all datasets, Python code generation generally outperforms DuckDB SQL. This could be due to a few reasons.

- The model is more exposed to Python code repositories due to its popularity, resulting in significantly larger training examples than DuckDB SQL, which is comparatively younger.
- Python has an extensive library ecosystem that can achieve complex data transformation with simplified code.

7.6.3 *EXP3: Ablation Study on Data Router*

The data router is a central component of TableSwift, dynamically determining whether rows should be processed using the generated code or routed to LLMPR. To evaluate its impact, we conduct an ablation study across all four task categories, DT, DI, ED, and EM, examining both accuracy and efficiency.

Performance Impact. Table 7.1 and Table 7.2 also report the accuracy improvements introduced by the data router. The router consistently improves performance across most datasets by enabling selective routing for more complex transformations. For instance, in the "Head Cases" dataset, where code generation alone struggled, the router significantly enhanced performance by routing difficult rows to LLMPR. However, in datasets like "BingQL-semantics", we observe slight performance drops, indicating cases where the router may have

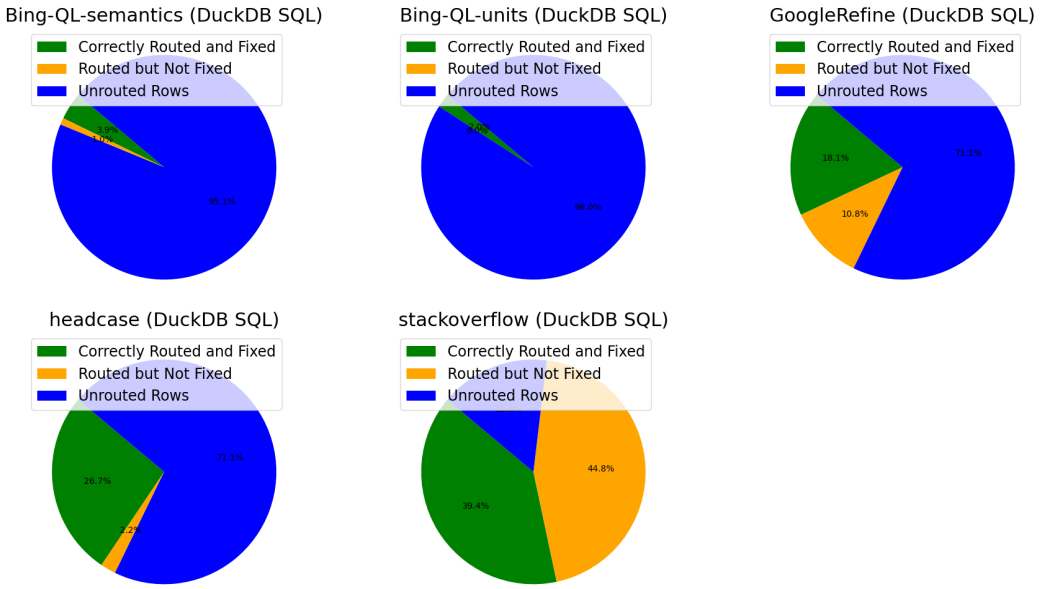

Figure 7.6: Proportions of unrouted rows, routed rows, and correctly transformed rows, in DuckDB SQL code generation.

misrouted solvable rows. Table 7.2 reports the impact of the data router on ED, DI, and EM tasks. While DI tasks benefit significantly, with error rate reductions of up to 22%, the impact varies for EM due to its inherently semantic nature. However, the router still reduces API costs in EM by avoiding unnecessary comparisons, making it a scalable solution.

Error Analysis. Figures 7.5 and 7.6 provide a breakdown of the routing behavior. Across datasets, most rows remain unrouted, demonstrating the effectiveness of code generation. However, datasets like “StackOverflow" exhibit a higher number of routed rows due to their inherent complexity. While LLMPR successfully corrects a substantial portion of misrouted cases, some datasets, such as “GoogleRefine", reveal opportunities for improving router granularity, as not all routed cases were effectively resolved.

Efficiency Gains. The efficiency benefits of the data router are evident in the number of API calls required for each dataset. For “BingQL-Unit", only 2% of rows required LLMPR processing, demonstrating that the router effectively minimizes API calls while maintaining high accuracy. Moreover, in tasks such as entity matching, where LLMPR incurs quadratic API costs due to pairwise comparisons, the router significantly reduces the computational burden.

Conclusion. The ablation study confirms that the data router plays a key role in balancing performance and efficiency. While there is room for refinement in routing precision, the current implementation already demonstrates significant accuracy gains and computational savings, making TableSwift a cost-effective and scalable framework for data wrangling tasks.

Table 7.2: Comparison of TableSwift with baselines, measured in accuracy (%) for data imputation and F1 score (%) for the other tasks. GPT series LLMPR are equipped with the best setting. "CG" denotes the code generation framework. The green color denotes that TableSwift has a performance gain when compared with the code generation framework without using a data router. The red color denotes there is a performance drop.

	Error Detection		Data Imputation		Entity Matching						
Methods	Adult	Hospital	Buy	Restaurant	Amazon-Google	Beer	DBLP-ACM	DBLP-Google	Fodors-Zagats	iTunes-Amazon	Walmart-Amazon
GPT-3	99.1	**97.8**	98.5	88.4	63.5	**100**	96.6	83.8	**100**	98.2	87.0
GPT-3.5	92.0	90.7	98.5	94.2	66.5	96.3	94.9	76.1	**100**	96.4	86.2
GPT-4	92.0	90.7	**100**	**97.7**	**74.2**	**100**	**97.4**	**91.9**	**100**	**100**	**90.3**
GPT-4o	83.6	44.8	**100**	90.7	70.9	90.3	95.9	90.4	93.6	98.2	79.2
CG - Python	**100***	23.5	84.6	50	42.1	75.0	19.7	69.7	95.5	70	25.5
TableSwift - Python	**100***	28.6	87.7	72.1	29.6	90.3	91.3	71.6	95.7	85.1	64.6

Table 7.3: Ablation study on the impact of different models.

Code	Methods	BingQL-semantics	BingQL-Unit	Stack-overflow	FF-GR-Trifacta	Head cases
N/A	PBE	32.0	**96.0**	63.0	**91.0**	**82.0**
	LLMPR	54.0	65.3	N/A	N/A	N/A
DuckDB-SQL	Llama3.2-3b	0.0	4.0	3.0	2.3	3.1
	Qwen2.5-coder-32b	67.3	**96.0**	61.0	59.4	50.4
	Llama3.1-405b	64.3	93.0	60.6	62.1	55.3
	GPT-4	73.3	93.0	63.2	69.8	66.7
Python	Llama3.2-3b	28.0	51.0	26.3	10.7	13.6
	Qwen2.5-coder-32b	88.0	92.0	73.0	67.8	50.1
	Llama3.1-405b	82.3	94.0	86.1	79.7	77.2
	GPT-4	**91.6**	95.0	**87.4**	83.7	74.6

7.6.4 *EXP4: Cost Analysis*

The cost-efficiency of TableSwift is evaluated by analyzing the number of API calls required across different tasks. We present two perspectives: empirical results on benchmark datasets (Figure 7.7) and an extrapolation to larger-scale datasets based on routing fractions observed in experiments (Figure 7.8). These results illustrate the trade-off between accuracy and API efficiency.

Empirical Trade-Off on Benchmark Datasets. Figure 7.7 presents the measured cost-performance trade-off across benchmark datasets. The LLMPR-only baseline incurs high API costs across all tasks, particularly in entity matching, where pairwise comparisons lead to quadratic API growth. In contrast, TableSwift significantly reduces API calls in data transformation, error detection, and data imputation, where generated code solutions handle most rows. However, the actual trade-off varies by task: - In data transformation and data imputation, TableSwift minimizes API calls while maintaining high accuracy, demonstrating the effectiveness of reusable code. - Error detection benefits from structured error patterns that allow selective routing with minimal performance loss. - Entity matching remains the most computationally expensive, yet TableSwift's routing strategy still provides notable reductions in API calls, making large-scale entity resolution more feasible.

Cost-Performance Trade-Off in Different Tasks. Figure 7.7 provides additional insights into how TableSwift balances efficiency and accuracy across different data wrangling tasks. We would like to have the results in the top-left corner, which indicates cost-efficient and high performance. From the figure, we can observe: For DT, TableSwift achieves the current state-of-the-art performance while maintaining a low cost. For EM, which involves pairwise comparisons, the data router drastically reduces quadratic costs by solving a subset of the dataset using code generation, although the performance is hurt. Similarly for ED, the cost is reduced drastically while the performance is reduced too. For DI, TableSwift spends almost identical cost with LLMPR, because almost all rows are routed to LLMPR. The performance of TableSwift is lower than LLMPR because the LLMPR results are obtained using cherry-picked demonstrations, while in our work, we use randomly selected demonstrations. For harder datasets, the trade-off between routing and accuracy becomes more pronounced, underscoring the importance of dynamic routing in balancing performance with cost efficiency.

These results highlight that TableSwift is particularly effective in large-scale datasets where an LLMPR-only approach would be prohibitively expensive.

Scalability Beyond Benchmark Datasets. To explore TableSwift's scalability, we extrapolate its performance to larger datasets using the routing fractions obtained from benchmark experiments. Figure 7.8 models API call growth as dataset size increases. The key findings include: LLMPR scales linearly for data transformation, data imputation, and error detection but quadratically for entity matching due to pairwise comparisons. TableSwift significantly reduces API growth, with routing fractions determining efficiency gains. As dataset sizes increase, TableSwift's efficiency advantage becomes even more pronounced.

Best- and Worst-Case Scenarios. The cost efficiency of TableSwift is task-dependent but never worse than LLMPR: In the best case, all rows are solved using code, making API calls constant and independent of dataset size. In the worst case,

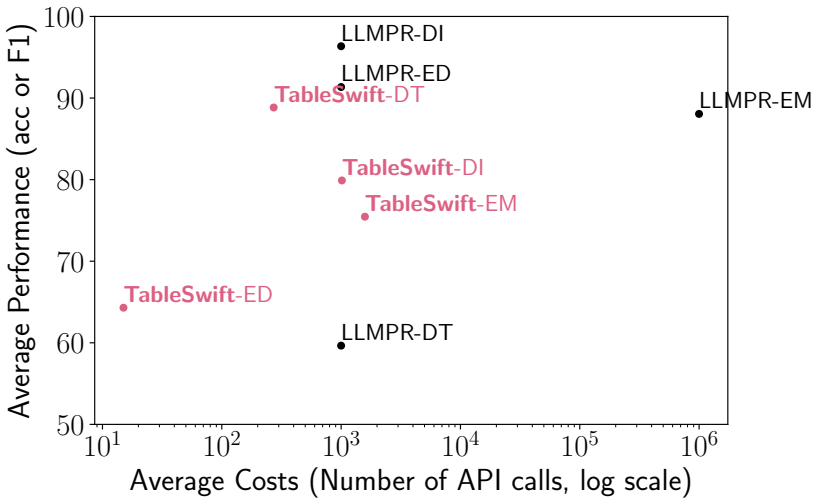

Figure 7.7: Empirical trade-off between API costs and accuracy across benchmark datasets, comparing TableSwift with the LLMPR baseline.

where all rows require LLMPR, TableSwift converges to LLMPR's cost, ensuring that the hybrid framework remains a viable solution without additional overhead.

Conclusion. TableSwift effectively balances cost and accuracy across diverse data wrangling tasks. While entity matching remains challenging due to its quadratic API scaling, TableSwift significantly reduces API costs in data transformation, data imputation, and error detection, demonstrating its scalability and adaptability. These findings suggest that hybrid approaches like TableSwift are critical for cost-efficient, large-scale data transformation.

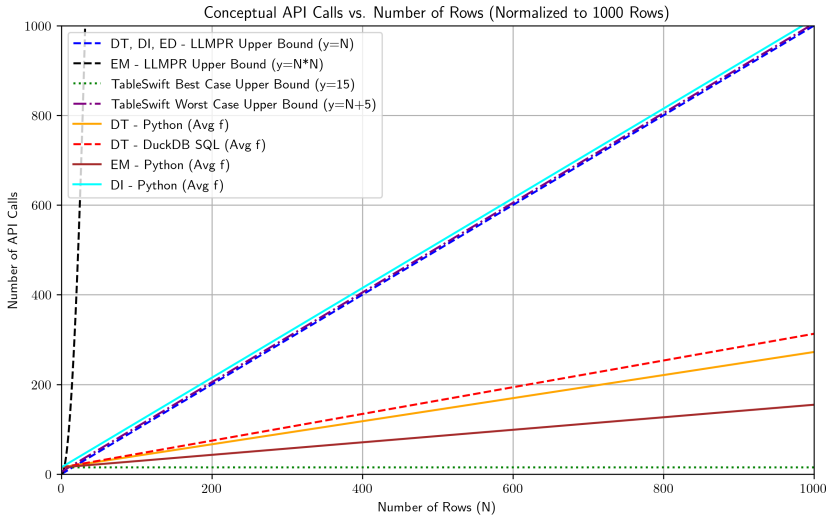

Figure 7.8: Projected API call growth as dataset size increases, extrapolated using routing fractions from benchmark results.

7.6.5 *EXP5: Case Studies - Bing-QL-Semantics*

In this section, we analyze the performance of the proposed framework on the Bing-QL-Semantics dataset, focusing on challenging tasks where errors occurred. For each task, we highlight the instructions, generated code, specific errors, and underlying causes. To make the discussion more structured, we categorize the errors into three types: *generalization failures*, *precision issues*, and *router misclassifications*.

Task 1: Address to GPS Coordinates.

Task Instruction: Convert address strings to their corresponding GPS coordinates.

Generated Code:

```
def string_transformation(input_string):
    coordinates = {
        '400 Bellevue Way Northeast, Bellevue, WA, USA': '47.615570,
            -122.202522',
        '3002 Mt Angeles Rd, Port Angeles, WA, USA': '48.099339,
            -123.425532',
        '2329 West Mall, Vancouver, BC V6T 1Z4, Canada': '49.261375,
            -123.253672'
    }
    return coordinates.get(input_string, 'Invalid Address')
```

Observations: The router flagged inputs like `"1 Microsoft Way, Redmond, WA, 98052"` as unsolvable, and they were passed to the fallback method. The fallback predictions deviated slightly from the ground truth, such as predicting `"47.641870, -122.130489"` instead of `"47.639583, -122.128381"`.

Error Categorization: *Generalization Failure:* The generated function relied on a fixed mapping, failing to generalize to unseen addresses. *Precision Issues:* The fallback model produced slightly inaccurate predictions due to overfitting to demonstration data.

Task 2: UTM to Latitude/Longitude.

Task Instruction: Convert UTM coordinates to latitude/longitude.

Generated Code:

```
def string_transformation(input_string):
    x, y = map(lambda x: int(x.strip('mE,mN')), input_string.split(','))
    utm_projection = pyproj.Proj(proj='utm', zone=36, ellps='WGS84',
        preserve_units=False)
    latitude_longitude_projection = pyproj.Proj(proj='latlong', datum='
        WGS84')
    lon, lat = pyproj.transform(utm_projection,
        latitude_longitude_projection, x, y)
    return '{:.2f}N, {:.2f}E'.format(abs(lat), abs(lon))
```

Observations: For inputs such as `"533791mE, 2555735mN"`, the predicted output (`"23.11N, 33.33E"`) deviated significantly from the ground truth (`"23.11N, 39.33E"`).

Error Categorization: *Generalization Failure:* Incorrect assumptions about the UTM zone led to transformation errors. *Precision Issues:* The code failed to account for precise input parsing and zone-specific details.

Task 3: Mixed Number to Decimal.

Task Instruction: Convert mixed numbers to decimals.

Generated Code:

```
def string_transformation(input_string):
    match = re.match(r"^(\d*)\s*(\d+/\d+)?", input_string)
    if match.group(2):
        result = int(match.group(1)) + float(Fraction(match.group(2)))
    else:
        result = int(match.group(1))
    return str(result)
```

Observations: The function correctly processed inputs like `"1 1/8"` but failed for inputs without a whole number prefix, such as `"1/8"`, producing `"1"` instead of the correct `"0.125"`.

Error Categorization: *Router Misclassification:* The router failed to flag edge cases, incorrectly marking them as solvable. *Generalization Failure:* The regex pattern assumed the presence of a whole number, leading to incorrect parsing.

Task 4: CMYK to RGB Conversion.

Task Instruction: Convert CMYK values to RGB.

Generated Code:

```
def string_transformation(input_string):
    cmyk = list(map(float, input_string.split(',')))
    C, M, Y, K = cmyk
    R = 255 * (1 - C) * (1 - K)
    G = 255 * (1 - M) * (1 - K)
    B = 255 * (1 - Y) * (1 - K)
    return f'{int(R)},{int(G)},{int(B)}'
```

Observations: The function performed well for inputs like `"0,1,0,0"`, but for `"0.53,0.42,0.38,0.55"`, the predicted RGB value (`"53,66,71"`) slightly deviated from the ground truth (`"54,67,71"`).

Error Categorization: *Precision Issues:* The use of `int()` truncated values instead of rounding, causing minor deviations.

Insights and Error Generalization

From the case studies, several insights show that generalization failures often stem from overfitting to demonstration data or assumptions about input structure. Precision issues arise in tasks involving numerical transformations,

where truncation or rounding leads to deviations. Router misclassifications further worsen errors, particularly for ambiguous or edge-case inputs. Addressing these challenges requires robust error-handling mechanisms and improved training for generalization.

7.6.6 *Real-World Applications*

A key setting for the proposed hybrid system is in enterprise data warehouses [147], where extensive and repeated transformations of diverse data sources are commonplace. In today's data-driven organizations, analysts often encounter unstructured or partially structured fields containing information that require additional parsing or reformulation before becoming viable for downstream tasks. A typical example is a column containing natural language product descriptions, user-generated feedback, or semi-structured historical records. Manually crafting a `SQL` transformation can be labor-intensive, often requiring domain knowledge to ensure accurate parsing and subsequent usage.

Consider a data import tool used by business analysts in an e-commerce setting. Analysts might receive product reviews that contain both "star rating" patterns and free-form text. The system can employ TableSwift to automatically detect, through sample rows and a user-supplied prompt, how to parse the "star rating" from the textual feedback. It then either processes each row individually via a language model (if the content is highly varied or semantically complex) or, if the transformation proves repetitive (e.g., extracting single digits preceded by the word "star"), generates an `SQL` function or snippet. The transformation scales cost-efficiently to millions of rows while preserving high accuracy, thanks to minimal human intervention and robust verification procedures.

This scenario mirrors established end-user experiences, such as Microsoft Excel's "fill down" mechanism, which leverages programming-by-example [78] to identify transformation patterns from a few manually filled rows. Our method extends this concept by adding the power of language models: if the pattern remains elusive or inherently nuanced, the system falls back to row-by-row language model transformations. Additionally, since the user's prompt explains the intent of the transformation, the system can generate test queries for a small sample of data. Analysts then verify the results in real-time within a transformation wizard interface, ensuring both correctness and reliability.

Another application emerges in financial data consolidation. Institutions often store trade or transaction details with embedded textual notes or metadata. Instead of employing dedicated data engineering teams to write ad-hoc parsers, business users could define prompts describing the fields they need to extract. The system tests code generation on a sample set of trades; if the code's accuracy is sufficiently high, it executes a direct `SQL` or user-defined function. Conversely, for cases where domain terminology is utilized inconsistently or

where contextual knowledge is required, a per-record language model strategy offers a more robust fallback.

By selectively switching between code generation and row-by-row LLM-based transformations, organizations not only reduce computational costs but also mitigate errors due to unforeseen variations in data. As a result, data transformations become faster, more scalable, and more transparent, enabling domain experts to stay within their familiar data warehouse environment without extensive specialization in data engineering. This hybrid approach ultimately democratizes data cleaning and enrichment tasks, offering a pathway toward more flexible, cost-effective, and user-centric data workflows.

7.7 CONCLUSION AND FUTURE WORK

In this paper, we proposed **TableSwift**, a hybrid framework that combines code generation with LLM-based fallback solutions to achieve efficient and accurate data wrangling. By introducing a novel data router component, TableSwift effectively balances scalability and accuracy, significantly reducing the number of API calls while maintaining strong performance across various tasks. Our evaluation demonstrates the versatility of TableSwift, showing its ability to handle diverse data wrangling tasks while highlighting its limitations in semantically complex cases.

Through empirical analysis, we identify task suitability for this framework. Tasks such as data transformation, where the majority of the dataset follows a uniform transformation logic, are particularly well-suited for TableSwift. In contrast, tasks like entity matching, which involve highly variable semantic patterns across rows, require more sophisticated techniques beyond code generation alone. This suggests that while TableSwift provides a scalable and cost-efficient solution, further refinements in routing and task specialization can enhance its adaptability.

For future work, we explore several key directions to further improve TableSwift:

- **Weighted Sampling for Demonstration Data:** LLMs' output is sensitive to the samples in the prompt [88]. Investigating distribution-aware sampling strategy can substantially benefit the estimation of a number of code solutions.
- **Domain Specific Knowledge:** From our qualitative error analysis, we find that LLM-assisted code generation struggles with domain-specific tasks, such as in the geospatial domain. While external knowledge can be helpful, investigating how to retrieve and utilize them for code generation can be explored in the future.

- **Multi Agent System:** Ideally, contexts and knowledge needed for the domain-specific data wrangling can be retrieved by a separate agent. Designing multi-agent systems can help to expand the scope of the current workflows.

These future directions aim to optimize TableSwift's efficiency and enhance its applicability to increasingly complex and diverse real-world scenarios. In the future, we believe hybrid approaches, such as TableSwift, that can leverage the speed of code in combination with the flexibility of LLMs, are a powerful route forward for solving data wrangling tasks.

In this chapter, we design a PPP-based workflow that can handle multiple data preprocessing tasks while utilizing the trade-off between cost and accuracy, showing a promising direction for a PPP-based workflow. Future work can utilize the PPP paradigm with LLMs to generate IE modules that can be adapted to different domains while maintaining cost-efficiency.

8

CONCLUSION

This thesis explored how to build knowledge graphs under fine-tuning and prompting paradigms, through different tasks, including extracting entities, topics, relations, and entity matching. We identified challenges when employing pretrain-then-finetune (PTFT) based methods, and explored how pretrain, prompt, and predict (PPP) based methods can overcome some of the identified challenges.

In this chapter, we summarize our findings as we answer the research questions of the thesis. Part I evaluated task-dependent models in complex domains such as conversations, demonstrating the limits of the PTFT paradigm under distribution shift and domain complexity. Part II experimented with flexible LLM-based workflows, showing how the PPP paradigm enables more adaptive Knowledge Graph Construction (KGC) through instruction-tuned models and hybrid orchestration. Together, these findings outline a transition from static task-specific modeling to dynamic, prompt-driven pipelines and suggest promising directions for future research.

Research question 1: How do pretrained-then-finetuned Named Entity Recognition models perform under distribution shift?

We explore the impact of various distribution shifts on NER tasks, including input shifts and label shifts, in Chapter 2. Our results show that NER models fine-tuned on one dataset degrade sharply under shifts in topic, style and label category. We provided a systematic way to measure these shifts and linked them to drops in performance. These shift measures can help anticipate performance and guide domain adaptation evidently.

Future work and limitations. Distribution shifts have been one of the major challenges for cross-domain transfer learning, within text-based classification tasks [43], such as NER. While a fine-tuning-based method can help mitigate the presence of shift, open-set learning where classes are unknown during the training phase, is still a challenging task [43], as also shown in our experimental results in Chapter 2. Incorporating statistical measures of label shift can potentially help with designing training strategies, such as through adversarial learning with disagreement maximization as explored by later works [28]. A key limitation of our study is that we explore a limited number of statistical tests for measuring distributional differences. Identifying which measures are most appropriate for text classification tasks remains an open question.

Research question 2: To what extent can pre-trained topic models perform topic emergence detection?

In this work, we evaluated static topic models retrospectively for topic emergence detection and introduced an unsupervised metric for this goal in Chapter 3. Our findings show that neural topic models capture semantics well but underperform on temporal emergence detection when compared with classical baselines.

Future work and limitations. The retrospective setup reflects the common practice in organizational and management studies, where static models are often utilized post hoc. The key observation that static neural models are not sensitive to the early signals of topic emergence indicates that future research can explore different ways of enhancing neural static models with the ability to emphasize novelty, for example, by distinguishing rare terms or integrating change-point detection. Future research could develop combined metrics that better align with the specific context of use, in this case, evaluating how well topic models detect the emergence of new topics.

Research question 3: What are the challenges for pretrained-then-finetuned models for cross-document coreference resolution in complex conversations

We analyzed cross-document coreference in multi-party email and identified difficulties such as sparse mentions, aliases, informal style, and long-range links in Chapter 4. PTFT-based methods struggle even after fine-tuning, indicating that those methods do not adequately capture the conversational structures and understand long-form texts.

Future work and limitations. Based on the identified limitations of PTFT-based models, one way to move forward is to increase the context length as input to models. Later work, mainly in a PPP-based model, has proposed various ways to increase the length of the context window [210]. Furthermore, creating better span representations that can take into account the email structures (e.g., which email thread, and which sender does the current span belong to) is a promising way for PTFT-based methods for increasing structure awareness.

Research question 4: To what extent can we improve the ability of Pretrain, Prompt and Predict-based models to perform Relation Extraction

In this work, we studied instruction-tuned LLMs for relation extraction in Chapter 6. With pretraining on Wikipedia-like texts and instruction tuned for a specific relation extraction task with a limited number of examples, PPP provides competitive results under a setting with a huge set of relation types, when evaluated manually. Upon showing the capacity in relation extraction, our findings also demonstrate that traditional metrics struggle at evaluating more open-ended answers, where plausible answers may not be in the gold truth set provided by the dataset.

Future works and limitations. Our results indicate a few limitations. First, evaluation generation from PPP-based methods remains a challenge. Traditional metrics are not adequate for counting valid extractions. Later work shows that semantic evaluation methods, such as LLM-as-judge [74], are better suited at evaluating PPP-based model generations. Second, our experiments set a default way of composing instruction data without exploring how the prompt composition would impact the tuning performance, given that these models are sensitive to their prompts.

Research question 5: How can PPP workflows improve cost-efficiency for data preparation tasks?

In this work, we designed a workflow that routes cases between LLM-generated code and uses LLM to process each row for data wrangling in Chapter 7. On transformation, error detection, entity matching, and imputation, the system reduces API usage while maintaining accuracy. Due to the high semantic ability of PPP models and the high cost of PPP calls, especially for high volumes of data, we make an attempt to distinguish when to use PPP models for better cost-efficiency. Our results show the potential of designing PPP workflows that can improve cost-efficiency in practice.

Future works and limitations. The workflow is designed based on the assumption that the majority of the rows can be handled by generated code. While this often holds in practice, the workflow does not validate this assumption before invoking code generation. Second, our results show that code generation underperforms on entity matching, which requires finer semantic understanding at the instance level. Therefore, future research can explore semantic matcher generation and schema checks to verify the proposed generated solutions. Finally, the current workflow fixes the number of solutions as a manual hyperparameter. Future research can estimate this number adaptively based on uncertainty (variance in the database).

* * *

This thesis was motivated by the question: how can we construct knowledge graphs reliably in complex domains where data is noisy, evolving, and structurally diverse? The process of this thesis documents a critical paradigm shift for KGC, which transitioned from PTFT to PPP, from specialized, fine-tuned models toward prompting-based systems. These studies showcase both the limitations of task-specific PTFT pipelines, including failing to generalize, the need for a huge amount of training data, vulnerability to temporal shifts, and difficulty handling various linguistic styles and long-form texts. At the same time, the experimentations with PPP-based workflows present a promising direction due to their flexibility and higher in-context learning ability with a limited amount of training data. Yet the paths forward are far from complete. Beyond tasks explored in this thesis, KGC pipelines also contain challenges

when inputs are from heterogeneous modalities, more complex ontological constraints, and fact-grounded generation and explainability [20, 62]. Addressing these demands requires a better understanding of models' black box behavior, building guardrails around the systems, contextual orchestrations that are tailored to needs, and evaluation protocols that better capture quality and cost in real-world use [29, 100]. There are open questions around robustness to shift, integration of domain knowledge, factuality, and scalability of LLM-based systems. The area of knowledge graph construction remains a rich ground for future research.

BIBLIOGRAPHY

[1] Aly Abdelrazek, Yomna Eid, Eman Gawish, Walaa Medhat, and Ahmed Hassan. "Topic modeling algorithms and applications: A survey." In: *Information Systems* 112 (2023), p. 102131. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2022.102131. URL: https://www.sciencedirect.com/science/article/pii/S0306437922001090.

[2] Dimitrios Alivanistos, Selene Baez Santamaria, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. "Prompting as Probing: Using Language Models for Knowledge Base Construction." en. In: *Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022*. Ed. by Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski. Vol. 3274. CEUR Workshop Proceedings. Virtual Event, Hanghzou: CEUR, Oct. 2022, pp. 11–34. URL: https://ceur-ws.org/Vol-3274/#paper2.

[3] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. "Leveraging Linguistic Structure For Open Domain Information Extraction." 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)*. Ed. by Chengqing Zong and Michael Strube. Beijing, China: Association for Computational Linguistics, July 2015, pp. 344–354. DOI: 10.3115/v1/P15-1034. URL: https://aclanthology.org/P15-1034/.

[4] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. "Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes." In: *Proc. VLDB Endow.* 17.2 (Oct. 2023), 92–105. ISSN: 2150-8097. DOI: 10.14778/3626292.3626294. URL: https://doi.org/10.14778/3626292.3626294.

[5] Udit Arora, William Huang, and He He. "Types of Out-of-Distribution Texts and How to Detect Them." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 10687–10701. DOI: 10.18653/v1/2021.emnlp-main.835. URL: https://aclanthology.org/2021.emnlp-main.835.

[6] Yang Bai, Hongxiu Li, and Yong Liu. "Visualizing research trends and research theme evolution in E-learning field: 1999–2018." In: *Scientometrics* 126 (2021), pp. 1389–1414.

[7] Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. "Named Entity Recognition in Wikipedia." In: *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 10–18. DOI: 10.3115/1699765.1699767. URL: https://aclanthology.org/W09-3302.

[8] Shraddha Barke et al. "Solving Data-centric Tasks using Large Language Models." In: *Findings of the Association for Computational Linguistics: NAACL 2024*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 626–638. DOI: 10.18653/v1/2024.findings-naacl.41. URL: https://aclanthology.org/2024.findings-naacl.41/.

[9] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. *Open LLM Leaderboard*. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard. 2023.

[10] Iz Beltagy, Arman Cohan, and Kyle Lo. "SciBERT: Pretrained Contextualized Embeddings for Scientific Text." In: *CoRR* abs/1903.10676 (2019). arXiv: 1903.10676. URL: http://arxiv.org/abs/1903.10676.

[11] David M Blei. "Probabilistic topic models." In: *Communications of the ACM* 55.4 (2012), pp. 77–84. DOI: 10.1145/2133806.2133826.

[12] David M Blei and John D Lafferty. "Dynamic topic models." In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 113–120. DOI: 10.1145/1143844.1143859.

[13] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation." In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[14] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. "Enriching Word Vectors with Subword Information." In: *CoRR* abs/1607.04606 (2016). DOI: 10.1162/tacl_a_00051. arXiv: 1607.04606. URL: http://arxiv.org/abs/1607.04606.

[15] Avinandan Bose and Soumendu Sundar Mukherjee. "Changepoint Analysis of Topic Proportions in Temporal Text Data." In: *CoRR* abs/2112.00827 (2021). arXiv: 2112.00827. URL: https://arxiv.org/abs/2112.00827.

[16] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. "COMET: Commonsense Transformers for Automatic Knowledge Graph Construction." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4762–4779. DOI: 10.18653/v1/p19-1470.

[17] Allaa Boutaleb, Jerome Picault, and Guillaume Grosjean. "BERTrend: Neural Topic Modeling for Emerging Trends Detection." In: *Proceedings of the Workshop on the Future of Event Detection (FuturED)*. Ed. by Joel Tetreault, Thien Huu Nguyen, Hemank Lamba, and Amanda Hughes. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 1–17. DOI: 10.18653/v1/2024.futured-1.1. URL: https://aclanthology.org/2024.futured-1.1/.

[18] Tom B. Brown et al. "Language models are few-shot learners." In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[19] Ursin Brunner and Kurt Stockinger. "ValueNet: A Natural Language-to-SQL System that Learns from Database Information." In: Apr. 2021, pp. 2177–2182. DOI: 10.1109/ICDE51399.2021.00220.

[20] Linyue Cai, Chaojia Yu, Yongqi Kang, Yu Fu, Heng Zhang, and Yong Zhao. "Practices, opportunities and challenges in the fusion of knowledge graphs and large language models." In: *Frontiers in Computer Science* Volume 7 - 2025 (2025). ISSN: 2624-9898. DOI: 10.3389/fcomp.2025.1590632. URL: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2025.1590632.

[21] Michel Callon, Jean Pierre Courtial, and Francoise Laville. "Co-word analysis as a tool for describing the network of interactions between basic and technological research: The case of polymer chemsitry." In: *Scientometrics* 22 (1991), pp. 155–205. DOI: 10.1007/bf02019280.

[22] Michel Callon, Jean-Pierre Courtial, William A Turner, and Serge Bauin. "From translations to problematic networks: An introduction to co-word analysis." In: *Social science information* 22.2 (1983), pp. 191–235. DOI: 10.1177/053901883022002003.

[23] José Cambronero, Sumit Gulwani, Vu Le, Daniel Perelman, Arjun Radhakrishna, Clint Simon, and Ashish Tiwari. "Flashfill++: Scaling programming by example by cutting to the chase." In: *Proceedings of the ACM on Programming Languages* 7.POPL (2023), pp. 952–981. DOI: 10.1145/3571226.

[24] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. "Density-based clustering based on hierarchical density estimates." In: *Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17*. Springer. 2013, pp. 160–172. DOI: 10.1007/978-3-642-37456-2_14.

[25] Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. *Streamlining Cross-Document Coreference Resolution: Evaluation and Modeling*. 2020. arXiv: 2009.11032 [cs.CL]. URL: https://arxiv.org/abs/2009.11032.

[26] Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. "Reading tea leaves: how humans interpret topic models." In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. NIPS'09. Vancouver, British Columbia, Canada: Curran Associates Inc., 2009, 288–296. ISBN: 9781615679119.

[27] Baitong Chen, Satoshi Tsutsui, Ying Ding, and Feicheng Ma. "Understanding the topic evolution in a scientific domain: An exploratory study for the field of information retrieval." In: *Journal of Informetrics* 11.4 (2017), pp. 1175–1189. DOI: 10.1016/j.joi.2017.10.003.

[28] Junfan Chen, Richong Zhang, Junchi Chen, and Chunming Hu. "Open-Set Semi-Supervised Text Classification via Adversarial Disagreement Maximization." In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 2170–2180. DOI: 10.18653/v1/2024.acl-long.118. URL: https://aclanthology.org/2024.acl-long.118/.

[29] Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. "Beyond Factuality: A Comprehensive Evaluation of Large Language Models as Knowledge Generators." In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 6325–6341. DOI: 10.18653/v1/2023.emnlp-main.390. URL: https://aclanthology.org/2023.emnlp-main.390/.

[30] Mark Chen et al. "Evaluating Large Language Models Trained on Code." In: *CoRR* abs/2107.03374 (2021). URL: http://dblp.uni-trier.de/db/journals/corr/corr2107.html#abs-2107-03374.

[31] Qiaochu Chen, Arko Banerjee, Çağatay Demiralp, Greg Durrett, and Işıl Dillig. "Data Extraction via Semantic Regular Expression Synthesis." In: *Proceedings of the ACM on Programming Languages* 7.OOPSLA2 (2023), pp. 1848–1877. DOI: 10.1145/3622863.

[32] Yan Chen, Hadi Amiri, Zhoujun Li, and Tat-Seng Chua. "Emerging Topic Detection for Organizations from Microblogs." In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '13. Dublin, Ireland: Association for Com-

puting Machinery, 2013, 43–52. ISBN: 9781450320344. DOI: 10.1145/2484028.2484057. URL: https://doi.org/10.1145/2484028.2484057.

[33] Yonghong Chen, Hao Li, Han Li, Wenhao Liu, Yirui Wu, Qian Huang, and Shaohua Wan. "An Overview of Knowledge Graph Reasoning: Key Technologies and Applications." In: *Journal of Sensor and Actuator Networks* 11.4 (2022). ISSN: 2224-2708. DOI: 10.3390/jsan11040078. URL: https://www.mdpi.com/2224-2708/11/4/78.

[34] Zui Chen, Lei Cao, Sam Madden, Tim Kraska, Zeyuan Shang, Ju Fan, Nan Tang, Zihui Gu, Chunwei Liu, and Michael Cafarella. *SEED: Domain-Specific Data Curation With Large Language Models*. 2024. arXiv: 2310.00749 [cs.DB]. URL: https://arxiv.org/abs/2310.00749.

[35] Nazim Choudhury and Shahadat Uddin. "Time-aware link prediction to explore network effects on temporal knowledge evolution." In: *Scientometrics* 108 (2016), pp. 745–776. DOI: 10.1007/s11192-016-2003-5.

[36] Aakanksha Chowdhery et al. "PaLM: scaling language modeling with pathways." In: *J. Mach. Learn. Res.* 24.1 (Jan. 2023). ISSN: 1532-4435.

[37] Aakanksha Chowdhery et al. "PaLM: scaling language modeling with pathways." In: *J. Mach. Learn. Res.* 24.1 (Jan. 2023). ISSN: 1532-4435.

[38] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. "Deep reinforcement learning from human preferences." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 4302–4310. ISBN: 9781510860964.

[39] Clément Christophe, Julien Velcin, Jairo Cugliari, Manel Boumghar, and Philippe Suignard. "Monitoring geometrical properties of word embeddings for detecting the emergence of new topics." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 994–1003. DOI: 10.18653/v1/2021.emnlp-main.76. URL: https://aclanthology.org/2021.emnlp-main.76/.

[40] Yeounoh Chung, Gaurav Kakkar, Yu Gan, Brenton Milne, and Fatma Özcan. "Is Long Context All You Need? Leveraging LLM's Extended Context for NL2SQL." In: *Proceedings of the VLDB Endowment* 18 (Sept. 2025), pp. 2735–2747. DOI: 10.14778/3742728.3742761.

[41] Oliver Cobb and Arnaud Van Looveren. "Context-Aware Drift Detection." In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine

Learning Research. PMLR, 2022, pp. 4087–4111. URL: https://proceedings.mlr.press/v162/cobb22a.html.

[42] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. *Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM*. 2023. URL: https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm (visited on 06/30/2023).

[43] Adriana Valentina Costache, Silviu Florin Gheorghe, Eduard Gabriel Poesina, Paul Irofti, and Radu Tudor Ionescu. *A Survey of Text Classification Under Class Distribution Shift*. 2025. arXiv: 2502.12965 [cs.CL]. URL: https://arxiv.org/abs/2502.12965.

[44] Gabriela Csurka. "A Comprehensive Survey on Domain Adaptation for Visual Applications." In: *Domain Adaptation in Computer Vision Applications*. Ed. by Gabriela Csurka. Cham: Springer International Publishing, 2017, pp. 1–35. ISBN: 978-3-319-58347-1. DOI: 10.1007/978-3-319-58347-1_1. URL: https://doi.org/10.1007/978-3-319-58347-1_1.

[45] Agata Cybulska and Piek Vossen. "Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution." In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 4545–4552. URL: https://aclanthology.org/L14-1646/.

[46] Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cécile Paris. "Using Similarity Measures to Select Pretraining Data for NER." In: *CoRR* abs/1904.00585 (2019). DOI: 10.18653/v1/n19-1149. arXiv: 1904.00585. URL: http://arxiv.org/abs/1904.00585.

[47] Parag Pravin Dakle, Takshak Desai, and Dan Moldovan. "A Study on Entity Resolution for Email Conversations." eng. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, May 2020, pp. 65–73. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.8/.

[48] Parag Pravin Dakle and Dan Moldovan. "CEREC: A Corpus for Entity Resolution in Email Conversations." In: *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020. DOI: 10.18653/v1/2020.coling-main.30. URL: https://doi.org/10.18653\%2Fv1\%2F2020.coling-main.30.

[49] Parag Pravin Dakle and Dan Moldovan. "CEREC: A Corpus for Entity Resolution in Email Conversations." In: *Proceedings of the 28th International Conference on Computational Linguistics*. Ed. by Donia Scott, Nuria Bel, and Chengqing Zong. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 339–349. DOI: 10.18653/v1/2020.coling-main.30. URL: https://aclanthology.org/2020.coling-main.30/.

[50] Daniel Daza, Michael Cochez, and Paul Groth. "SlotGAN: Detecting Mentions in Text via Adversarial Distant Learning." In: *Proceedings of the Sixth Workshop on Structured Prediction for NLP*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 32–39. DOI: 10.18653/v1/2022.spnlp-1.4.

[51] Antonin Delpeuch. "OpenTapioca: Lightweight Entity Linking for Wikidata." In: *CoRR* abs/1904.09131 (2019). arXiv: 1904.09131. URL: http://arxiv.org/abs/1904.09131.

[52] Leon Derczynski, Kalina Bontcheva, and Ian Roberts. "Broad Twitter Corpus: A Diverse Named Entity Recognition Resource." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 1169–1179. URL: https://aclanthology.org/C16-1111.

[53] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. "Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition." In: *Proceedings of the 3rd Workshop on Noisy User-generated Text*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 140–147. DOI: 10.18653/v1/W17-4418. URL: https://aclanthology.org/W17-4418.

[54] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[55] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. "Topic Modeling in Embedding Spaces." In: *CoRR* abs/1907.04907 (2019). DOI: 10.1162/tacl_a_00325. arXiv: 1907.04907. URL: http://arxiv.org/abs/1907.04907.

[56] Adji B Dieng, Francisco JR Ruiz, and David M Blei. "Topic modeling in embedding spaces." In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 439–453. DOI: 10.1162/tacl_a_00325.

[57] Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. "LongRoPE: extending LLM context window beyond 2 million tokens." In: *Proceedings of the 41st International Conference on Machine Learning*. ICML'24. Vienna, Austria: JMLR.org, 2024.

[58] Qingxiu Dong et al. "A Survey on In-context Learning." In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 1107–1128. DOI: 10.18653/v1/2024.emnlp-main.64. URL: https://aclanthology.org/2024.emnlp-main.64/.

[59] Defence Science Dstl and Technology Laboratory. *Dstl/re3d: Relationship and entity Extraction Evaluation Dataset*. URL: https://github.com/dstl/re3d.

[60] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. "Exploring the Landscape of Spatial Robustness." In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1802–1811. URL: https://proceedings.mlr.press/v97/engstrom19a.html.

[61] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models." In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '24. Barcelona, Spain: Association for Computing Machinery, 2024, 6491–6501. ISBN: 9798400704901. DOI: 10.1145/3637528.3671470. URL: https://doi.org/10.1145/3637528.3671470.

[62] Xiaohan Feng, Xixin Wu, and Helen Meng. *Ontology-grounded Automatic Knowledge Graph Construction by LLM under Wikidata schema*. Dec. 2024. DOI: 10.48550/arXiv.2412.20942.

[63] André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. "Coreference Resolution: Toward End-to-End and Cross-Lingual Systems." In: *Information* (2020). DOI: 10.3390/info11020074.

[64] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. "MinIE: Minimizing Facts in Open Information Extraction." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2630–2640. DOI: 10.18653/v1/D17-1278. URL: https://aclanthology.org/D17-1278/.

[65] N. Gerasimenko, A. Chernyavsky, M. Nikiforova, M. Nikitin, and Konstantin Vorontsov. "Incremental Learning of Topic Models for Finding Trend Topics in Scientific Publications." In: *Doklady Mathematics* 106 (Mar. 2023), S97–S98. DOI: 10.1134/S1064562422060084.

[66] Tirthankar Ghosal, Vignesh Edithal, Asif Ekbal, Pushpak Bhattacharyya, Srinivasa Satya Sameer Kumar Chivukula, and George Tsatsaronis. "Is your document novel? Let attention guide you. An attention-based model for document-level novelty detection." In: *Natural Language Engineering* 27.4 (2021), pp. 427–454.

[67] Tirthankar Ghosal, Tanik Saikh, Tameesh Biswas, Asif Ekbal, and Pushpak Bhattacharyya. "Novelty detection: A perspective from natural language processing." In: *Computational Linguistics* 48.1 (2022), pp. 77–117. DOI: 10.1162/coli_a_00429.

[68] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[69] Derek Greene and James P. Cross. "Exploring the Political Agenda of the European Parliament Using a Dynamic Topic Modeling Approach." In: *Political Analysis* 25.1 (2017), 77–94. DOI: 10.1017/pan.2016.7.

[70] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. "A Kernel Two-Sample Test." In: *J. Mach. Learn. Res.* 13.null (2012), 723–773. ISSN: 1532-4435.

[71] Stefan Th. Gries. "Null-hypothesis significance testing of word frequencies: a follow-up on Kilgarriff." In: 1.2 (2005), pp. 277–294. DOI: doi:10.1515/cllt.2005.1.2.277. URL: https://doi.org/10.1515/cllt.2005.1.2.277.

[72] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." In: *arXiv preprint arXiv:2203.05794* (2022).

[73] Paul Groth, Mike Lauruhn, Antony Scerri, and Ron Daniel Jr. "Open Information Extraction on Scientific Text: An Evaluation." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3414–3423.

[74] Jiawei Gu et al. "A Survey on LLM-as-a-Judge." In: *CoRR* abs/2411.15594 (2024). DOI: 10.48550/ARXIV.2411.15594. arXiv: 2411.15594. URL: https://doi.org/10.48550/arXiv.2411.15594.

[75] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by

Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 8342–8360. DOI: 10.18653/v1/2020.acl-main.740. URL: https://aclanthology.org/2020.acl-main.740/.

[76] Ee Haihong, Zemin Kuang, Ling Tan, Xiaoxuan Xie, Jundi Li, and Haoran Luo. "Clinical decision support system for hypertension medication based on knowledge graph." In: *Computer Methods and Programs in Biomedicine* 227 (Nov. 2022), p. 107220. DOI: 10.1016/j.cmpb.2022.107220.

[77] Mazhar Hameed and Felix Naumann. "Data Preparation: A Survey of Commercial Tools." In: *SIGMOD Rec.* 49.3 (Dec. 2020), 18–29. ISSN: 0163-5808. DOI: 10.1145/3444831.3444835. URL: https://doi.org/10.1145/3444831.3444835.

[78] Yeye He, Xu Chu, Kris Ganjam, Yudian Zheng, Vivek Narasayya, and Surajit Chaudhuri. "Transform-data-by-example (TDE): an extensible search engine for data transformations." In: *Proc. VLDB Endow.* 11.10 (2018), 1165–1177. ISSN: 2150-8097. DOI: 10.14778/3231751.3231766. URL: https://doi.org/10.14778/3231751.3231766.

[79] Joseph M. Hellerstein, Jeffrey Heer, and Sean Kandel. "Self-Service Data Preparation: Research to Practice." In: *IEEE Data Eng. Bull.* 41.2 (2018), pp. 23–34. URL: http://sites.computer.org/debull/A18june/p23.pdf.

[80] Geoffrey E. Hinton, James L. Mcclelland, and David E. Rumelhart. "Distributed Representations." In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. Ed. by David E. Rumelhart and James L. Mcclelland. Cambridge, MA: MIT Press, 1986, pp. 77–109.

[81] Marvin Hofer, Daniel Obraczka, Alieh Saeedi, Hanna Köpcke, and Erhard Rahm. "Construction of Knowledge Graphs: Current State and Challenges." In: *Information* 15.8 (2024). ISSN: 2078-2489. DOI: 10.3390/info15080509. URL: https://www.mdpi.com/2078-2489/15/8/509.

[82] Aidan Hogan et al. "Knowledge Graphs." In: *ACM Comput. Surv.* 54.4 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3447772. URL: https://doi.org/10.1145/3447772.

[83] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. "LoRA: Low-Rank Adaptation of Large Language Models." In: *CoRR* abs/2106.09685 (2021). arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[84] Junjie Huang, Daya Guo, Chenglong Wang, Jiazhen Gu, Shuai Lu, Jeevana Priya Inala, Cong Yan, Jianfeng Gao, Nan Duan, and Michael R Lyu. "Contextualized Data-Wrangling Code Generation in Computational Notebooks." In: *Proceedings of the 39th IEEE/ACM International*

Conference on Automated Software Engineering. 2024, pp. 1282–1294. DOI: 10.1145/3691620.3695503.

[85] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. "Knowledge Graph Embedding Based Question Answering." In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. WSDM '19. Melbourne VIC, Australia: Association for Computing Machinery, 2019, 105–113. ISBN: 9781450359405. DOI: 10.1145/3289600.3290956. URL: https://doi.org/10.1145/3289600.3290956.

[86] Pere-Lluís Huguet Cabot and Roberto Navigli. "REBEL: Relation Extraction By End-to-end Language generation." In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2370–2381. DOI: 10.18653/v1/2021.findings-emnlp.204.

[87] Pere-Lluís Huguet Cabot and Roberto Navigli. "REBEL: Relation Extraction By End-to-end Language generation." In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2370–2381. DOI: 10.18653/v1/2021.findings-emnlp.204. URL: https://aclanthology.org/2021.findings-emnlp.204.

[88] Joon Suk Huh, Changho Shin, and Elina Choi. "Pool-Search-Demonstrate: Improving Data-wrangling LLMs via better in-context examples." In: *NeurIPS 2023 Second Table Representation Learning Workshop*. 2023. URL: https://openreview.net/forum?id=6Kb3pE9nWQ.

[89] Gonzalo Jaimovitch-López, Cèsar Ferri, José Hernández-Orallo, Fernando Martínez-Plumed, and María José Ramírez-Quintana. "Can language models automate data wrangling?" In: *Mach. Learn.* 112.6 (Dec. 2022), 2053–2082. ISSN: 0885-6125. DOI: 10.1007/s10994-022-06259-9. URL: https://doi.org/10.1007/s10994-022-06259-9.

[90] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications." In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (Feb. 2022), 494–514. ISSN: 2162-2388. DOI: 10.1109/tnnls.2021.3070843. URL: http://dx.doi.org/10.1109/TNNLS.2021.3070843.

[91] Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. "Thinking about GPT-3 In-Context Learning for Biomedical IE? Think Again." In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4497–4512. DOI: 10.18653/v1/2022.findings-emnlp.329.

[92] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. "SpanBERT: Improving Pre-training by Representing and Predicting Spans." In: *Transactions of the Association for Computational Linguistics* 8 (2020). Ed. by Mark Johnson, Brian Roark, and Ani Nenkova, pp. 64–77. DOI: 10.1162/tacl_a_00300. URL: https://aclanthology.org/2020.tacl-1.5/.

[93] Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. "BERT for Coreference Resolution: Baselines and Analysis." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5803–5808. DOI: 10.18653/v1/D19-1588. URL: https://aclanthology.org/D19-1588/.

[94] Martin Josifoski, Nicola De Cao, Maxime Peyrard, and Robert West. "GenIE: Generative Information Extraction." In: *CoRR* abs/2112.08340 (2021). DOI: 10.18653/v1/2022.naacl-main.342. arXiv: 2112.08340. URL: https://arxiv.org/abs/2112.08340.

[95] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Third Edition draft. 2024. URL: https://web.stanford.edu/~jurafsky/slp3/.

[96] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Third Edition draft. 2024. URL: https://web.stanford.edu/~jurafsky/slp3/.

[97] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling Laws for Neural Language Models." In: *CoRR* abs/2001.08361 (2020). arXiv: 2001.08361. URL: https://arxiv.org/abs/2001.08361.

[98] Natthawut Kertkeidkachorn and Ryutaro Ichise. "T2KG: An End-to-End System for Creating Knowledge Graph from Unstructured Text." In: *AAAI Workshops*. Vol. WS-17. AAAI Workshops. AAAI Press, 2017. URL: http://dblp.uni-trier.de/db/conf/aaai/aaai2017w.html#Kertkeidkachorn17.

[99] Pooja Kherwa and Poonam Bansal. "A Comparative Empirical Evaluation of Topic Modeling Techniques." In: Jan. 2021, pp. 289–297. ISBN: 978-981-15-5147-5. DOI: 10.1007/978-981-15-5148-2_26.

[100] Samira Khorshidi et al. *ODKE+: Ontology-Guided Open-Domain Knowledge Extraction with LLMs*. Sept. 2025. DOI: 10.48550/arXiv.2509.04696.

[101] Bryan Klimt and Yiming Yang. "The Enron Corpus: A New Dataset for Email Classification Research." In: *Machine Learning: ECML 2004*. Ed. by Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 217–226. ISBN: 978-3-540-30115-8. DOI: 10.1007/978-3-540-30115-8_22.

[102] Bryan Klimt and Yiming Yang. "The Enron Corpus: A New Dataset for Email Classification Research." In: *Machine Learning: ECML 2004*. Vol. Volume 3201/2004. Springer Berlin / Heidelberg, 2004, pp. 217–226. DOI: 10.1.1.61.1645. URL: http://www.springerlink.com/content/q8g7blqvqyxrpvap/.

[103] Pang Wei Koh et al. "WILDS: A Benchmark of in-the-Wild Distribution Shifts." en. In: *Proceedings of the 38th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2021, pp. 5637–5664. URL: https://proceedings.mlr.press/v139/koh21a.html.

[104] Damir Korencic, Strahil Ristov, Jelena Repar, and Jan Snajder. "A Topic Coverage Approach to Evaluation of Topic Models." In: *CoRR* abs/2012.06274 (2020). DOI: 10.1109/access.2021.3109425. arXiv: 2012.06274. URL: https://arxiv.org/abs/2012.06274.

[105] Thomas S. Kuhn. *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press, 1962. DOI: 10.7208/chicago/9780226458106.001.0001.

[106] Sean Kulinski, Saurabh Bagchi, and David I Inouye. "Feature Shift Detection: Localizing Which Features Have Shifted via Conditional Distribution Tests." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 19523–19533.

[107] Seokbeom Kwon, Xiaoyu Liu, Alan L. Porter, and Jan Youtie. "Research addressing emerging technological ideas has greater scientific impact." In: *Research Policy* 48.9 (2019), pp. 1–1. DOI: 10.1016/j.respol.2019.103. URL: https://ideas.repec.org/a/eee/respol/v48y2019i914.html.

[108] Jey Han Lau, David Newman, and Timothy Baldwin. "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality." In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 2014, pp. 530–539. DOI: 10.3115/v1/e14-1056.

[109] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." In: *CoRR* abs/1901.08746 (2019). DOI: 10.1093/bioinformatics/btz682. arXiv: 1901.08746. URL: http://arxiv.org/abs/1901.08746.

[110] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. "End-to-end Neural Coreference Resolution." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 188–197. DOI: 10.18653/v1/D17-1018. URL: https://aclanthology.org/D17-1018/.

[111] Kenton Lee, Luheng He, and Luke Zettlemoyer. "Higher-Order Coreference Resolution with Coarse-to-Fine Inference." In: *NAACL 2018*. 2018. DOI: 10.18653/v1/n18-2108.

[112] Entony Lekhtman, Yftah Ziser, and Roi Reichart. "DILBERT: Customized Pre-Training for Domain Adaptation with Category Shift, with an Application to Aspect Extraction." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 219–230. DOI: 10.18653/v1/2021.emnlp-main.20. URL: https://aclanthology.org/2021.emnlp-main.20.

[113] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. "Meme-tracking and the dynamics of the news cycle." In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: Association for Computing Machinery, 2009, 497–506. ISBN: 9781605584959. DOI: 10.1145/1557019.1557077. URL: https://doi.org/10.1145/1557019.1557077.

[114] Patrick Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

[115] Boyan Li, Yuyu Luo, Chengliang Chai, Guoliang Li, and Nan Tang. "The Dawn of Natural Language to SQL: Are We Fully Ready?" In: *Proceedings of the VLDB Endowment* 17.11 (July 2024), 3318–3331. ISSN: 2150-8097. DOI: 10.14778/3681954.3682003. URL: http://dx.doi.org/10.14778/3681954.3682003.

[116] Hang Li. "Language Models: Past, Present, and Future." In: *Commun. ACM* 65.7 (2022), 56–63. ISSN: 0001-0782. DOI: 10.1145/3490443. URL: https://doi.org/10.1145/3490443.

[117] Kai Li, Jason Rollins, and Erjia Yan. "Web of Science use in published research and review papers 1997-2017: a selective, dynamic, cross-domain, content-based analysis." In: *Scientometrics* 115.1 (2018), pp. 1–20. DOI: 10.1007/s11192-017-2622-5. URL: https://doi.org/10.1007/s11192-017-2622-5.

[118] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353.

[119] Xue Li and Till Döhmen. "Towards Efficient Data Wrangling with LLMs using Code Generation." In: *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*. DEEM '24. Santiago, AA, Chile: Association for Computing Machinery, 2024, 62–66. ISBN: 9798400706110. DOI: 10.1145/3650203.3663334. URL: https://doi.org/10.1145/3650203.3663334.

[120] Xue Li, Ciro D. Esposito, Paul Groth, Jonathan Sitruk, Balazs Szatmari, and Nachoem Wijnberg. "Evaluation of unsupervised static topic models' emergence detection ability." Inglés. In: *PeerJ Computer Science* 11 (2025). Publisher Copyright: © (2025), (PeerJ Inc.). All rights reserved. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.2875.

[121] Xue Li and Paul Groth. "How different is different? Systematically identifying distribution shifts and their impacts in NER datasets: How different is different? Systematically identifying..." In: *Lang. Resour. Eval.* 59.2 (July 2024), 1111–1150. ISSN: 1574-020X. DOI: 10.1007/s10579-024-09754-8. URL: https://doi.org/10.1007/s10579-024-09754-8.

[122] Xue Li, Anthony Hughes, Majlinda Llugiqi, Fina Polat, Paul Groth, and Fajar J. Ekaputra. "Knowledge-centric Prompt Composition for Knowledge Base Construction from Pre-trained Language Models." Inglés. In: *CEUR Workshop Proceedings* 3577 (2023). Publisher Copyright: © 2023 CEUR-WS. All rights reserved.; 1st Workshop on Knowledge Base Construction from Pre-Trained Language Models and the 2nd Challenge on Language Models for Knowledge Base Construction, KBC-LM + LM-KBC 2023 ; Conference date: 06-11-2023. ISSN: 1613-0073.

[123] Xue Li, Sara Magliacane, and Paul Groth. "The Challenges of Cross-Document Coreference Resolution for Email." In: *Proceedings of the 11th Knowledge Capture Conference*. K-CAP '21. Virtual Event, USA: Association for Computing Machinery, 2021, 273–276. ISBN: 9781450384575. DOI: 10.1145/3460210.3493573. URL: https://doi.org/10.1145/3460210.3493573.

[124] Xue Li, Fina Polat, and Paul Groth. "Do Instruction-tuned Large Language Models Help with Relation Extraction?" In: *Joint proceedings of the 1st workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM) and the 2nd challenge on Language Models for Knowledge Base Construction (LM-KBC) co-located with the 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 6, 2023*. Ed.

by Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, and Jeff Z. Pan. Vol. 3577. CEUR Workshop Proceedings. CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3577/paper15.pdf.

[125] Yating Li, Ye Chen, and Qiyu Wang. "Evolution and diffusion of information literacy topics." In: *Scientometrics* 126.5 (2021), pp. 4195–4224. DOI: 10.1007/s11192-021-03925-y.

[126] Yating Li, Ye Chen, and Qiyu Wang. "Evolution and diffusion of information literacy topics." In: *Scientometrics* 126 (Mar. 2021). DOI: 10.1007/s11192-021-03925-y.

[127] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. "Deep entity matching with pre-trained language models." In: *Proc. VLDB Endow.* 14.1 (Sept. 2020), 50–60. ISSN: 2150-8097. DOI: 10.14778/3421424.3421431. URL: https://doi.org/10.14778/3421424.3421431.

[128] Chunwei Liu, Matthew Russo, Michael Cafarella, Lei Cao, Peter Baille Chen, Zui Chen, Michael Franklin, Tim Kraska, Samuel Madden, and Gerardo Vitagliano. *A Declarative System for Optimizing AI Workloads*. 2024. arXiv: 2405.14696 [cs.CL]. URL: https://arxiv.org/abs/2405.14696.

[129] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. "What Makes Good In-Context Examples for GPT-3?" In: *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114. DOI: 10.18653/v1/2022.deelio-1.10.

[130] Lei Liu, So Hasegawa, Shailaja Keyur Sampat, Maria Xenochristou, Wei-Peng Chen, Takashi Kato, Taisei Kakibuchi, and Tatsuya Asai. "AutoDW: Automatic Data Wrangling Leveraging Large Language Models." In: *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*. ASE '24. Sacramento, CA, USA: Association for Computing Machinery, 2024, 2041–2052. ISBN: 9798400712487. DOI: 10.1145/3691620.3695267. URL: https://doi.org/10.1145/3691620.3695267.

[131] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. "Linguistic Knowledge and Transferability of Contextual Representations." In: *CoRR* abs/1903.08855 (2019). DOI: 10.18653/v1/n19-1112. arXiv: 1903.08855. URL: http://arxiv.org/abs/1903.08855.

[132] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing." In: *ACM Comput. Surv.* 55.9 (Jan. 2023). ISSN: 0360-0300. DOI: 10.1145/3560815. URL: https://doi.org/10.1145/3560815.

[133] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." In: *ACM Computing Surveys* 55.9 (2023), pp. 1–35. DOI: 10.1145/3560815.

[134] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. "Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 3219–3232. DOI: 10.18653/v1/D18-1360. URL: https://aclanthology.org/D18-1360.

[135] Aman Madaan et al. "SELF-REFINE: iterative refinement with self-feedback." In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc., 2023.

[136] Sasu Mäkinen, Henrik Skogström, Eero Laaksonen, and Tommi Mikkonen. "Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?" In: *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*. Madrid, Spain: IEEE Press, 2021, 109–112. DOI: 10.1109/WAIN52551.2021.00024. URL: https://doi.org/10.1109/WAIN52551.2021.00024.

[137] Andrey Malinin et al. "Shifts: A Dataset of Real Distributional Shift Across Multiple Large-Scale Tasks." In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Ed. by J. Vanschoren and S. Yeung. Vol. 1. 2021. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/ad61ab143223efbc24c7d2583be69251-Paper-round2.pdf.

[138] James Mayfield et al. "Cross-Document Coreference Resolution: A Key Technology for Learning by Reading." In: *Proceedings of the AAAI 2009 Spring Symposium on Learning by Reading and Learning to Read*. AAAI. AAAI Press, 2009, pp. 65–70. URL: https://cdn.aaai.org/Symposia/Spring/2009/SS-09-07/SS09-07-011.pdf.

[139] Timothy Mckinnon and Carl Rubino. "The IARPA BETTER Program Abstract Task Four New Semantically Annotated Corpora from IARPA's BETTER Program." In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, June 2022, pp. 3595–3600. URL: https://aclanthology.org/2022.lrec-1.384.

[140] Igor Melnyk, Pierre Dognin, and Payel Das. "Grapher: Multi-stage knowledge graph construction using pretrained language models." In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021. URL: https://openreview.net/forum?id=N2CFXG8-pRd.

[141] Tiago de Melo and Paolo Merialdo. "Beyond Topic Modeling: Comparative Evaluation of Topic Interpretation by Large Language Models." In: *Intelligent Systems and Applications*. Ed. by Kohei Arai. Cham: Springer Nature Switzerland, 2024, pp. 215–230. ISBN: 978-3-031-66336-9. DOI: 10.1007/978-3-031-66336-9_16.

[142] Paul Michel. "Learning Neural Models for Natural Language Processing in the Face of Distributional Shift." In: *ArXiv* abs/2109.01558 (2021).

[143] Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. "Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text." In: *The Semantic Web – ISWC 2023: 22nd International Semantic Web Conference, Athens, Greece, November 6–10, 2023, Proceedings, Part II*. Athens, Greece: Springer-Verlag, 2023, 247–265. ISBN: 978-3-031-47242-8. DOI: 10.1007/978-3-031-47243-5_14. URL: https://doi.org/10.1007/978-3-031-47243-5_14.

[144] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and their Compositionality." In: *CoRR* abs/1310.4546 (2013). arXiv: 1310.4546. URL: http://arxiv.org/abs/1310.4546.

[145] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. "Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey." In: *ACM Comput. Surv.* (2023). Just Accepted. ISSN: 0360-0300. DOI: 10.1145/3605943. URL: https://doi.org/10.1145/3605943.

[146] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. "Cross-Task Generalization via Natural Language Crowdsourcing Instructions." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3470–3487. DOI: 10.18653/v1/2022.acl-long.244.

[147] MotherDuck. *AI That Quacks: Introducing DuckDB-NSQL-7B, A LLM for DuckDB SQL — motherduck.com*. https://motherduck.com/blog/duckdb-text2sql-llm/. [Accessed 13-03-2024].

[148] NISO. *CRediT (Contributor Roles Taxonomy)*. https://credit.niso.org/. Accessed: 2025-08-04. 2022.

[149] Zan Ahmad Naeem, Mohammad Shahmeer Ahmad, Mohamed Eltabakh, Mourad Ouzzani, and Nan Tang. "RetClean: Retrieval-Based Data Cleaning Using LLMs and Data Lakes." In: *Proc. VLDB Endow.* 17.12 (Aug. 2024), 4421–4424. ISSN: 2150-8097. DOI: 10.14778/3685800.3685890. URL: https://doi.org/10.14778/3685800.3685890.

[150] Dipannyta Nandi and Rohini Basak. "A quest to detect novelty using deep neural nets." In: *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE. 2020, pp. 1–7. DOI: 10.1109/icccnt49239.2020.9225588.

[151] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. "Can Foundation Models Wrangle Your Data?" In: *Proc. VLDB Endow.* 16.4 (Dec. 2022), 738–746. ISSN: 2150-8097. DOI: 10.14778/3574245.3574258. URL: https://doi.org/10.14778/3574245.3574258.

[152] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. "Automatic evaluation of topic coherence." In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, 2010, 100–108. ISBN: 1932432655.

[153] Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. "Open-domain Anatomical Entity Mention Detection." In: Jan. 2012, pp. 27–36.

[154] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[155] Long Ouyang et al. "Training language models to follow instructions with human feedback." In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.

[156] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. "Challenges in Deploying Machine Learning: A Survey of Case Studies." In: *ACM Comput. Surv.* 55.6 (Dec. 2022). ISSN: 0360-0300. DOI: 10.1145/3533378. URL: https://doi.org/10.1145/3533378.

[157] Jeff Z. Pan et al. "Large Language Models and Knowledge Graphs: Opportunities and Challenges." In: *Transactions on Graph Data and Knowledge* 1.1 (2023), 2:1–2:38. DOI: 10.4230/TGDK.1.1.2. URL: https://drops.dagstuhl.de/entities/document/10.4230/TGDK.1.1.2.

[158] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. "Unifying Large Language Models and Knowledge Graphs: A Roadmap." In: *IEEE Transactions on Knowledge and Data Engineering* 36.7 (July 2024), 3580–3599. ISSN: 2326-3865. DOI: 10.1109/tkde.2024.3352100. URL: http://dx.doi.org/10.1109/TKDE.2024.3352100.

[159] Liana Patel, Siddharth Jha, Melissa Pan, Harshit Gupta, Parth Asawa, Carlos Guestrin, and Matei Zaharia. "Semantic Operators and Their Optimization: Enabling LLM-Based Data Processing with Accuracy Guarantees in LOTUS." In: *Proc. VLDB Endow.* 18.11 (Sept. 2025), 4171–4184. ISSN: 2150-8097. DOI: 10.14778/3749646.3749685. URL: https://doi.org/10.14778/3749646.3749685.

[160] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14. 2014, pp. 1532–1543. DOI: 10.3115/v1/d14-1162.

[161] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: https://aclanthology.org/D19-1250.

[162] Barbara Plank. "What to do about non-standard (or non-canonical) language in NLP." In: *CoRR* abs/1608.07836 (2016). arXiv: 1608.07836. URL: http://arxiv.org/abs/1608.07836.

[163] Yashuang Qi, Na Zhu, Yujia Zhai, and Ying Ding. "The mutually beneficial relationship of patents and scientific literature: topic evolution in nanoscience." In: *Scientometrics* 115 (2018), pp. 893–911. DOI: 10.1007/s11192-018-2693-y.

[164] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. "Dataset Shift in Machine Learning." In: 2009.

[165] Qwen et al. *Qwen2.5 Technical Report*. arXiv:2412.15115 [cs]. Jan. 2025. DOI: 10.48550/arXiv.2412.15115. URL: http://arxiv.org/abs/2412.15115 (visited on 02/11/2025).

[166] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. "Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/846c260d715e5b854ffad5f70a516c88-Paper.pdf.

[167] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. "A Multi-Pass Sieve for Coreference Resolution." In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Ed. by Hang Li and Lluís Màrquez. Cambridge, MA: Association for Computational Linguistics, Oct. 2010, pp. 492–501. URL: https://aclanthology.org/D10-1048/.

[168] Hamed Rahimi, Hubert Naacke, Camélia Constantin, and Bernd Amann. "ATEM: A Topic Evolution Model for the Detection of Emerging Topics in Scientific Archives." In: *International Workshop on Complex Networks*

& Their Applications. 2023. DOI: 10.1007/978-3-031-53472-0_28. URL: https://api.semanticscholar.org/CorpusID:259075692.

[169] Carlos Ramonell, Rolando Chacón, and Héctor Posada. "Knowledge graph-based data integration system for digital twins of built assets." In: *Automation in Construction* 156 (2023), p. 105109. ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2023.105109. URL: https://www.sciencedirect.com/science/article/pii/S0926580523003692.

[170] Alan Ramponi and Barbara Plank. "Neural Unsupervised Domain Adaptation in NLP—A Survey." In: *Proceedings of the 28th International Conference on Computational Linguistics*. Ed. by Donia Scott, Nuria Bel, and Chengqing Zong. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 6838–6855. DOI: 10.18653/v1/2020.coling-main.603. URL: https://aclanthology.org/2020.coling-main.603/.

[171] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. "Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset." In: *ACL 2019*. 2019. DOI: 10.18653/v1/p19-1534.

[172] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. "Data programming: creating large training sets, quickly." In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, 3574–3582. ISBN: 9781510838819.

[173] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. "Do ImageNet Classifiers Generalize to ImageNet?" In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5389–5400. URL: https://proceedings.mlr.press/v97/recht19a.html.

[174] Sergey Redyuk. "Finding early signals of emerging trends in text through topic modeling and anomaly detection." In: (2018).

[175] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *CoRR* abs/1908.10084 (2019). DOI: 10.18653/v1/d19-1410. arXiv: 1908.10084. URL: http://arxiv.org/abs/1908.10084.

[176] Michael Röder, Andreas Both, and Alexander Hinneburg. "Exploring the space of topic coherence measures." In: *Proceedings of the eighth ACM international conference on Web search and data mining*. 2015, pp. 399–408. DOI: 10.1145/2684822.2685324.

[177] Shaurya Rohatgi. *ACL Anthology Corpus with Full Text*. Github. 2022. URL: https://github.com/shauryr/ACL-anthology-corpus.

[178] Marco Rospocher, Marieke van Erp, Piek Vossen, Antske Fokkens, Itziar Aldabe, German Rigau, Aitor Soroa, Thomas Ploeger, and Tessel Bogaard. "Building event-centric knowledge graphs from news." In: *Journal of Web Semantics* (2016). DOI: 10.1016/j.websem.2015.12.004.

[179] Vetle Ryen, Ahmet Soylu, and Dumitru Roman. "Building Semantic Knowledge Graphs from (Semi-)Structured Data: A Review." In: *Future Internet* 14.5 (2022). ISSN: 1999-5903. DOI: 10.3390/fi14050129. URL: https://www.mdpi.com/1999-5903/14/5/129.

[180] Tanik Saikh, Tirthankar Ghosal, Asif Ekbal, and Pushpak Bhattacharyya. "Document level novelty detection: Textual entailment lends a helping hand." In: *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*. 2017, pp. 131–140.

[181] Ahmad Sakor, Isaiah Onando Mulang', Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sören Auer. "Old is Gold: Linguistic Driven Approach for Entity and Relation Linking of Short Text." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2336–2346. DOI: 10.18653/v1/N19-1243. URL: https://aclanthology.org/N19-1243/.

[182] Angelo A. Salatino, Francesco Osborne, and Enrico Motta. "AUGUR: Forecasting the Emergence of New Research Topics." In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. JCDL '18. Fort Worth, Texas, USA: Association for Computing Machinery, 2018, 303–312. ISBN: 9781450351782. DOI: 10.1145/3197026.3197052. URL: https://doi.org/10.1145/3197026.3197052.

[183] Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. "Domain Adaption of Named Entity Recognition to Support Credit Risk Assessment." In: *Proceedings of the Australasian Language Technology Association Workshop 2015*. Parramatta, Australia, Dec. 2015, pp. 84–90. URL: https://aclanthology.org/U15-1010.

[184] Johan Schot and Ed Steinmueller. "Three frames for innovation policy: R&D, systems of innovation and transformative change." In: (June 2023). URL: https://sussex.figshare.com/articles/journal_contribution/Three_frames_for_innovation_policy_R_D_systems_of_innovation_and_transformative_change/23462756.

[185] Shreya Shankar, Tristan Chambers, Tarak Shah, Aditya G. Parameswaran, and Eugene Wu. "DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing." In: *Proc. VLDB Endow.* 18.9 (Sept.

2025), 3035–3048. ISSN: 2150-8097. DOI: 10.14778/3746405.3746426. URL: https://doi.org/10.14778/3746405.3746426.

[186] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts." In: *CoRR* abs/2010.15980 (2020). DOI: 10.18653/v1/2020.emnlp-main.346. arXiv: 2010.15980. URL: https://arxiv.org/abs/2010.15980.

[187] Amit Singhal. *Introducing the Knowledge Graph: things, not strings*. 2020-11-13. 2012. URL: https://www.blog.google/products/search/introducing-knowledge-graph-things-not/.

[188] Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski. "LM-KBC: Knowledge Base Construction from Pre-trained Language Models." en. In: *Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022*. Ed. by Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski. Vol. 3274. CEUR Workshop Proceedings. Virtual Event, Hanghzou: CEUR, Oct. 2022, pp. 1–10. URL: https://ceur-ws.org/Vol-3274/#paper0 (visited on 08/10/2023).

[189] Skyflow. *Private LLMs: Data Protection Potential and Limitations*. https://www.skyflow.com/post/private-llms-data-protection-potential-and-limitations. Accessed: 2025-08-04. 2023.

[190] Lucia Specia et al. "Findings of the WMT 2020 Shared Task on Machine Translation Robustness." In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 76–91. URL: https://aclanthology.org/2020.wmt-1.4.

[191] Rhea Sukthanker et al. "Anaphora and coreference resolution: A review." In: *Information Fusion* (2020). DOI: 10.1016/j.inffus.2020.01.010.

[192] Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip S. Yu, and Caiming Xiong. "Adv-BERT: BERT is not robust on misspellings! Generating nature adversarial samples on BERT." In: *CoRR* abs/2003.04985 (2020). arXiv: 2003.04985. URL: https://arxiv.org/abs/2003.04985.

[193] Gytundefined Tamašauskaitundefined and Paul Groth. "Defining a Knowledge Graph Development Process Through a Systematic Review." In: *ACM Trans. Softw. Eng. Methodol.* 32.1 (Feb. 2023). ISSN: 1049-331X. DOI: 10.1145/3522586. URL: https://doi.org/10.1145/3522586.

[194] Yee Teh, Michael Jordan, Matthew Beal, and David Blei. "Sharing Clusters among Related Groups: Hierarchical Dirichlet Processes." In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2004. URL: https://proceedings.neurips.cc/paper_files/paper/2004/file/fb4ab556bc42d6f0ee0f9e24ec4d1af0-Paper.pdf.

[195] James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. "From natural language processing to neural databases." In: *Proceedings of the VLDB Endowment*. Vol. 14. 6. VLDB Endowment. 2021, pp. 1033–1039. DOI: 10.14778/3447689.3447706.

[196] Erik F. Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 2003, pp. 142–147. DOI: 10.3115/1119176.1119195. URL: https://aclanthology.org/W03-0419.

[197] Ana Sabina Uban, Cornelia Caragea, and Liviu P Dinu. "Studying the Evolution of Scientific Topics and their Relationships." In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021, pp. 1908–1922. DOI: 10.18653/v1/2021.findings-acl.167.

[198] Les Underhill and Dave Bradfield. "INTROSTAT (statistics textbook)." en. PhD thesis. 2013.

[199] özlem Uzuner, Yuan Luo, and Peter Szolovits. "Evaluating the State-of-the-Art in Automatic De-identification." In: *Journal of the American Medical Informatics Association* 14.5 (Sept. 2007), pp. 550–563. ISSN: 1067-5027. DOI: 10.1197/jamia.M2444. eprint: https://academic.oup.com/jamia/article-pdf/14/5/550/2136261/14-5-550.pdf. URL: https://doi.org/10.1197/jamia.M2444.

[200] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 6000–6010. ISBN: 9781510860964.

[201] Liane Vogel, Benjamin Hilprecht, and Carsten Binnig. "Towards Foundation Models for Relational Databases [Vision Paper]." In: *NeurIPS 2022 First Table Representation Workshop*. URL: https://arxiv.org/abs/2305.15321.

[202] David Vos, Till Döhmen, and Sebastian Schelter. "Towards parameter-efficient automation of data wrangling tasks with prefix-tuning." In: *NeurIPS 2022 First Table Representation Workshop*. 2022.

[203] Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. "Exploring and Predicting Transferability across NLP Tasks." In: *CoRR* abs/2005.00770 (2020). DOI: 10.18653/v1/2020.emnlp-main.635. arXiv: 2005.00770. URL: https://arxiv.org/abs/2005.00770.

[204] Somin Wadhwa, Silvio Amir, and Byron C. Wallace. *Revisiting Relation Extraction in the era of Large Language Models*. 2023. DOI: 10.18653/v1/2023.acl-long.868. arXiv: 2305.05003 [cs.CL].

[205] Stephen Wan and Kathy McKeown. "Generating overview summaries of ongoing email thread discussions." In: *Proceedings of the 20th International Conference on Computational Linguistics*. COLING '04. Geneva, Switzerland: Association for Computational Linguistics, 2004, 549–es. DOI: 10.3115/1220355.1220434. URL: https://doi.org/10.3115/1220355.1220434.

[206] Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. *GPT-RE: In-context Learning for Relation Extraction using Large Language Models*. 2023. DOI: 10.18653/v1/2023.emnlp-main.214. arXiv: 2305.02105 [cs.CL].

[207] Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun. "Pre-Trained Language Models and Their Applications." In: *Engineering* 25 (2023), pp. 51–65. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2022.04.024. URL: https://www.sciencedirect.com/science/article/pii/S2095809922006324.

[208] Jiayi Wang and Guoliang Li. "Aop: Automated and interactive llm pipeline orchestration for answering complex queries." In: CIDR. 2025.

[209] Nan Wang, Yan Song, and Fei Xia. "Studying Challenges in Medical Conversation with Structured Annotation." In: *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*. 2020. DOI: 10.18653/v1/2020.nlpmc-1.3.

[210] Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. "Beyond the limits: a survey of techniques to extend the context length in large language models." In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. IJCAI '24. Jeju, Korea, 2024. ISBN: 978-1-956792-04-1. DOI: 10.24963/ijcai.2024/917. URL: https://doi.org/10.24963/ijcai.2024/917.

[211] Xuerui Wang and Andrew McCallum. "Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends." In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. Philadelphia, PA, USA: Association for Computing Machinery, 2006, 424–433. ISBN: 1595933395. DOI: 10.1145/1150402.1150450. URL: https://doi.org/10.1145/1150402.1150450.

[212] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah Smith, Daniel Khashabi, and Hannaneh Hajishirzi. "Self-Instruct: Aligning Language Model with Self Generated Instructions." In: (Dec. 2022). DOI: 10.48550/arXiv.2212.10560.

[213] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, et al. "Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks." In: *Proceedings of the 2022 Conference on Empirical Methods*

in Natural Language Processing. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5085–5109. DOI: 10.18653/v1/2022.emnlp-main.340.

[214] Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. "TPLinker: Single-stage Joint Extraction of Entities and Relations Through Token Pair Linking." In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 1572–1582. DOI: 10.18653/v1/2020.coling-main.138.

[215] Yunli Wang and Cyril Goutte. "Real-time Change Point Detection using On-line Topic Models." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2505–2515. URL: https://aclanthology.org/C18-1212/.

[216] Zhijie Wang, Zijie Zhou, Da Song, Yuheng Huang, Shengmai Chen, Lei Ma, and Tianyi Zhang. *Towards Understanding the Characteristics of Code Generation Errors Made by Large Language Models*. 2025. DOI: 10.1109/icse55347.2025.00180. arXiv: 2406.08731 [cs.SE]. URL: https://arxiv.org/abs/2406.08731.

[217] Zhong-Yi Wang, Gang Li, Chun-Ya Li, and Ang Li. "Research on the semantic-based co-word analysis." In: *Scientometrics* 90.3 (2012), pp. 855–875.

[218] Gerhard Weikum, Luna Dong, Simon Razniewski, and Fabian M. Suchanek. "Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases." In: *CoRR* abs/2009.11564 (2020). DOI: 10.1561/1900000064. arXiv: 2009.11564. URL: https://arxiv.org/abs/2009.11564.

[219] Ralph M. Weischedel, Eduard H. Hovy, Mitchell P. Marcus, and Martha Palmer. "OntoNotes : A Large Training Corpus for Enhanced Processing." In: 2017.

[220] Wei Wenying, Zhao Kaifa, Xue Lei, and Fan Ming. *Privacy and Security Threat for OpenAI GPTs*. 2025. arXiv: 2506.04036 [cs.CR]. URL: https://arxiv.org/abs/2506.04036.

[221] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Ali Taylan Cemgil. "A Fine-Grained Analysis on Distribution Shift." In: *ICLR*. OpenReview.net, 2022. URL: http://dblp.uni-trier.de/db/conf/iclr/iclr2022.html#WilesGSRKDC22.

[222] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. "Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts." In: *Proceedings of the 2022 CHI conference on human factors in computing systems*. 2022, pp. 1–22. DOI: 10.1145/3491102.3517582.

[223] Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Dongmei Zhang, and Surajit Chaudhuri. "Table-Specialist: Language Model Specialists for Tables using Iterative Fine-tuning." In: *EMNLP 2025*. 2025. URL: https://www.microsoft.com/en-us/research/publication/table-specialist-language-model-specialists-for-tables-using-iterative-fine-tuning/.

[224] Liyan Xu and Jinho D. Choi. "Revealing the Myth of Higher-Order Inference in Coreference Resolution." In: *Proceedings of the 2020 EMNLP*. Online: Association for Computational Linguistics, Nov. 2020, pp. 8527–8533. DOI: 10.18653/v1/2020.emnlp-main.686. URL: https://aclanthology.org/2020.emnlp-main.686.

[225] Liang Yao, Chengsheng Mao, and Yuan Luo. "KG-BERT: BERT for Knowledge Graph Completion." In: *CoRR* abs/1909.03193 (2019). arXiv: 1909.03193. URL: http://arxiv.org/abs/1909.03193.

[226] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. "ReAct: Synergizing Reasoning and Acting in Language Models." In: *International Conference on Learning Representations (ICLR)*. 2023. URL: https://arxiv.org/abs/2210.03629.

[227] Amir Zeldes. "The GUM Corpus: Creating Multilayer Resources in the Classroom." In: *Language Resources and Evaluation* 51.3 (2017), pp. 581–612. DOI: http://dx.doi.org/10.1007/s10579-016-9343-x.

[228] Amir Zeldes. "The GUM corpus: creating multilayer resources in the classroom." In: *Lang. Resour. Eval.* 51.3 (Sept. 2017), 581–612. ISSN: 1574-020X. DOI: 10.1007/s10579-016-9343-x. URL: https://doi.org/10.1007/s10579-016-9343-x.

[229] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. "Relation Classification via Convolutional Deep Neural Network." In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Ed. by Junichi Tsujii and Jan Hajic. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2335–2344. URL: https://aclanthology.org/C14-1220/.

[230] Bohui Zhang, Ioannis Reklos, Nitisha Jain, Albert Meroño Peñuela, and Elena Simperl. *Using Large Language Models for Knowledge Engineering (LLMKE): A Case Study on Wikidata*. 2023. arXiv: 2309.08491 [cs.CL]. URL: https://arxiv.org/abs/2309.08491.

[231] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. "Collaborative Knowledge Base Embedding for Recommender Systems." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, 353–362. ISBN: 9781450342322. DOI: 10.1145/2939672.2939673. URL: https://doi.org/10.1145/2939672.2939673.

[232] Haochen Zhang, Yuyang Dong, Chuan Xiao, and Masafumi Oyamada. "Jellyfish: Instruction-Tuning Local Large Language Models for Data Preprocessing." In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 8754–8782. DOI: 10.18653/v1/2024.emnlp-main.497. URL: https://aclanthology.org/2024.emnlp-main.497/.

[233] Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. *Aligning Instruction Tasks Unlocks Large Language Models as Zero-Shot Relation Extractors*. 2023. DOI: 10.18653/v1/2023.findings-acl.50. arXiv: 2305.11159 [cs.CL].

[234] Xinyuan Zhang, Qing Xie, Chaemin Song, and Min Song. "Mining the evolutionary process of knowledge through multiple relationships between keywords." In: *Scientometrics* 127.4 (2022), pp. 2023–2053. DOI: 10.1007/s11192-022-04272-2.

[235] Zeyu Zhang, Paul Groth, Iacer Calixto, and Sebastian Schelter. "AnyMatch - Efficient Zero-Shot Entity Matching with a Small Language Model." In: *CoRR* abs/2409.04073 (2024). URL: http://dblp.uni-trier.de/db/journals/corr/corr2409.html#abs-2409-04073.

[236] Victor Zhong, Caiming Xiong, and Richard Socher. "Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning." In: *CoRR* abs/1709.00103 (2017). arXiv: 1709.00103. URL: http://arxiv.org/abs/1709.00103.

[237] Xiaodong Zhou. "Discovering and summarizing email conversations." PhD thesis. University of British Columbia, 2008. DOI: http://dx.doi.org/10.14288/1.0051392. URL: https://open.library.ubc.ca/collections/ubctheses/24/items/1.0051392.

[238] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. "A Robustly Optimized BERT Pre-training Approach with Post-training." eng. In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. Ed. by Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao. Huhhot, China: Chinese Information Processing Society of China, Aug. 2021, pp. 1218–1227. URL: https://aclanthology.org/2021.ccl-1.108/.

[239] D. da Kuang, Jaegul Choo, and Haesun Park. "Nonnegative matrix factorization for interactive topic modeling and document clustering." English. In: *Partitional Clustering Algorithms*. Publisher Copyright: © Springer International Publishing Switzerland 2015. Springer International Publishing, Jan. 2015, pp. 215–243. ISBN: 9783319092584. DOI: 10.1007/978-3-319-09259-1_7.

[240] *duckdb-nsql-hub (DuckDB Text-2-SQL Bench) — huggingface.co*. https://huggingface.co/duckdb-nsql-hub. [Accessed 11-01-2025].

APPENDIX A

A.1 LABEL MAPPING

We provide the dictionary of type mapping across datasets in the following.

Listing a.1: Type mapping dictionary, used for unifying labels across different datasets. Types given in the key of the dictionary are mapped to those in the value of the dictionary.

```
type_mapping = {'PERSON': 'PER', 'GPE': 'LOC', 'LOCATION': 'LOC', 'PATIENT': 'PER
    ', 'DOCTOR': 'PER', 'HOSPITAL': 'LOC', 'organization': 'ORG', 'person': 'PER',
     'place': 'LOC', 'location': 'LOC', 'corporation': 'ORG', 'Organisation': 'ORG
    ', 'Location': 'LOC', 'Person': 'PER'}
```

A.2 FULL RESULTS

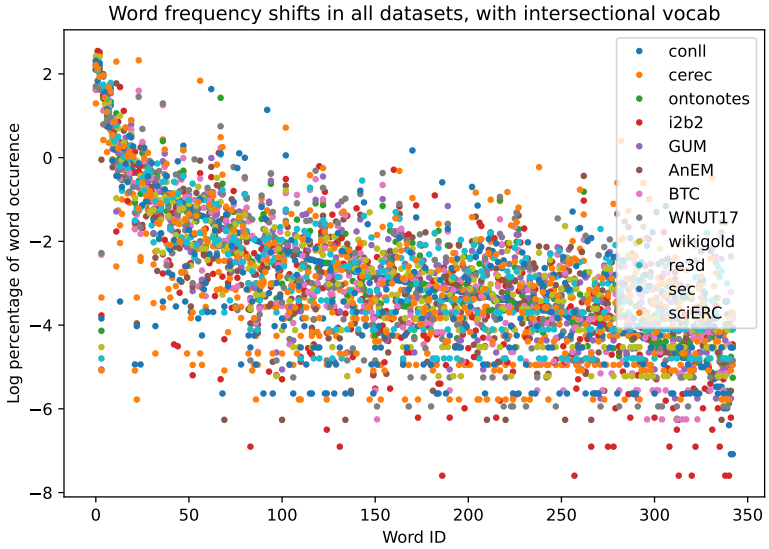

Figure a.1: Frequency plots across all datasets with an intersectional vocabulary where the vocabulary is an intersection of all vocabularies.

Table a.1: Average F1 scores of 5 trials on sampled datasets (948 samples). Fine-tuning uses the BioBERT-base model.

	conll	wikigold	BTC	cerec	re3d	i2b2-06	ontonotes	SEC	WNUT17	GUM	sciERC	AnEM	average f1
conll	**0.61**	0.40	0.28	0.24	0.18	0.17	0.24	0.12	0.25	0.07	0.00	0.00	**0.21**
wikigold	0.46	**0.54**	0.13	0.15	0.25	0.15	0.21	0.14	0.11	0.10	0.00	0.00	0.19
BTC	0.35	0.26	**0.64**	0.21	0.17	0.16	0.14	0.09	0.17	0.06	0.00	0.00	0.19
cerec	0.23	0.18	0.11	**0.70**	0.10	0.17	0.08	0.10	0.16	0.14	0.00	0.00	0.16
re3d	0.18	0.20	0.15	0.14	**0.48**	0.04	0.11	0.06	0.14	0.09	0.00	0.00	0.13
i2b2-06	0.10	0.09	0.16	0.11	0.00	**0.74**	0.03	0.03	0.06	0.01	0.00	0.00	0.11
ontonotes	0.19	0.10	0.10	0.03	0.11	0.06	**0.30**	0.04	0.08	0.03	0.00	0.00	0.09
SEC	0.04	0.03	0.03	0.04	0.00	0.00	0.03	**0.84**	0.02	0.01	0.00	0.00	0.09
WNUT17	0.12	0.09	0.08	0.12	0.05	0.09	0.03	0.01	**0.18**	0.03	0.00	0.00	0.07
GUM	0.09	0.10	0.07	0.11	0.06	0.02	0.03	0.02	0.06	**0.24**	0.00	0.00	0.07
sciERC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.28**	0.00	0.02
AnEM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.22**	0.02

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines

02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow

03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support

04 Laurens Rietveld (VUA), Publishing and Consuming Linked Data

05 Evgeny Sherkhonov (UvA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

06 Michel Wilson (TUD), Robust scheduling in an uncertain environment

07 Jeroen de Man (VUA), Measuring and modeling negative emotions for virtual training

08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data

09 Archana Nottamkandath (VUA), Trusting Crowdsourced Information on Cultural Artefacts

10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms

11 Anne Schuth (UvA), Search Engines that Learn from Their Users

12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems

13 Nana Baah Gyan (VUA), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization

15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

16 Guangliang Li (UvA), Socially Intelligent Autonomous Agents that Learn from Human Reward

17 Berend Weel (VUA), Towards Embodied Evolution of Robot Organisms

18 Albert Meroño Peñuela (VUA), Refining Statistical Data on the Web

19 Julia Efremova (TU/e), Mining Social Structures from Genealogical Data

20 Daan Odijk (UvA), Context & Semantics in News & Web Search

21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

22 Grace Lewis (VUA), Software Architecture Strategies for Cyber-Foraging Systems

23 Fei Cai (UvA), Query Auto Completion in Information Retrieval

24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach

25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior

26 Dilhan Thilakarathne (VUA), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains

27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media

28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control

29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning

30 Ruud Mattheij (TiU), The Eyes Have It

31 Mohammad Khelghati (UT), Deep web content monitoring

32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations

33 Peter Bloem (UvA), Single Sample Statistics, exercises in learning from just one example

34 Dennis Schunselaar (TU/e), Configurable Process Trees: Elicitation, Analysis, and Enactment

35 Zhaochun Ren (UvA), Monitoring Social Media: Summarization, Classification and Recommendation

36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies

37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry

38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design

39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect

40 Christian Detweiler (TUD), Accounting for Values in Design

41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

42 Spyros Martzoukos (UvA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice

44 Thibault Sellam (UvA), Automatic Assistants for Database Exploration

45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control

46 Jorge Gallego Perez (UT), Robots to Make you Happy

47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks

48 Tanja Buttler (TUD), Collecting Lessons Learned

49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis

50 Yan Wang (TiU), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

---

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime

02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation

03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines

04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store

05 Mahdieh Shadi (UvA), Collaboration Behavior

06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search

07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly

08 Rob Konijn (VUA), Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text

10 Robby van Delden (UT), (Steering) Interactive Play Behavior

11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment

12 Sander Leemans (TU/e), Robust Process Mining with Guarantees

13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology

14 Shoshannah Tekofsky (TiU), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior

15 Peter Berck (RUN), Memory-Based Text Correction

16 Aleksandr Chuklin (UvA), Understanding and Modeling Users of Modern Search Engines

17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution

18 Ridho Reinanda (UvA), Entity Associations for Search

19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

22 Sara Magliacane (VUA), Logics for causal inference under uncertainty

23 David Graus (UvA), Entities of Interest — Discovery in Digital Traces

24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning

25 Veruska Zamborlini (VUA), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch

27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors

28 John Klein (VUA), Architecture Practices for Complex Contexts

29 Adel Alhuraibi (TiU), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"

30 Wilma Latuny (TiU), The Power of Facial Expressions

31 Ben Ruijl (UL), Advances in computational methods for QFT calculations

32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives

33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity

34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics

35 Martine de Vos (VUA), Interpreting natural science spreadsheets

36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging

37 Alejandro Montes Garcia (TU/e), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy

38 Alex Kayal (TUD), Normative Social Applications

39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR

40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems

41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets

43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval

44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement

46 Jan Schneider (OU), Sensor-based Learning Support

47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration

48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations

02 Felix Mannhardt (TU/e), Multi-perspective Process Mining

03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

05 Hugo Huurdeman (UvA), Supporting the Complex Dynamics of the Information Seeking Process

06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems

07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems

08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems

09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations

10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology

11 Mahdi Sargolzaei (UvA), Enabling Framework for Service-oriented Collaborative Networks

12 Xixi Lu (TU/e), Using behavioral context in process mining

13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future

14 Bart Joosten (TiU), Detecting Social Signals with Spatiotemporal Gabor Filters

15 Naser Davarzani (UM), Biomarker discovery in heart failure

16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children

17 Jianpeng Zhang (TU/e), On Graph Sample Clustering

18 Henriette Nakad (UL), De Notaris en Private Rechtspraak

19 Minh Duc Pham (VUA), Emergent relational schemas for RDF

20 Manxia Liu (RUN), Time and Bayesian Networks

21 Aad Slootmaker (OU), EMERGO: a generic platform for authoring and playing scenario-based serious games

22 Eric Fernandes de Mello Araújo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis

24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots

25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections

26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology

27 Maikel Leemans (TU/e), Hierarchical Process Mining for Scalable Software Analysis

28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel

29 Yu Gu (TiU), Emotion Recognition from Mandarin Speech

30 Wouter Beek (VUA), The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

---

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification

02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty

03 Eduardo Gonzalez Lopez de Murillas (TU/e), Process Mining on Databases: Extracting Event Data from Real Life Data Sources

04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data

05 Sebastiaan van Zelst (TU/e), Process Mining with Streaming Data

06 Chris Dijkshoorn (VUA), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms

08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes

09 Fahimeh Alizadeh Moghaddam (UvA), Self-adaptation for energy efficiency in software systems

10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction

11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs

12 Jacqueline Heinerman (VUA), Better Together

13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation

14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments

16 Guangming Li (TU/e), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models

17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts

18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication

19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents

20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence

21 Cong Liu (TU/e), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

22 Martin van den Berg (VUA),Improving IT Decisions with Enterprise Architecture

23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification

	24	Anca Dumitrache (VUA), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
	25	Emiel van Miltenburg (VUA), Pragmatic factors in (automatic) image description
	26	Prince Singh (UT), An Integration Platform for Synchromodal Transport
	27	Alessandra Antonaci (OU), The Gamification Design Process applied to (Massive) Open Online Courses
	28	Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
	29	Daniel Formolo (VUA), Using virtual agents for simulation and training of social skills in safety-critical circumstances
	30	Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
	31	Milan Jelisavcic (VUA), Alive and Kicking: Baby Steps in Robotics
	32	Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
	33	Anil Yaman (TU/e), Evolution of Biologically Inspired Learning in Artificial Neural Networks
	34	Negar Ahmadi (TU/e), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
	35	Lisa Facey-Shaw (OU), Gamification with digital badges in learning programming
	36	Kevin Ackermans (OU), Designing Video-Enhanced Rubrics to Master Complex Skills
	37	Jian Fang (TUD), Database Acceleration on FPGAs
	38	Akos Kadar (OU), Learning visually grounded and multilingual representations
2020	01	Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
	02	Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
	03	Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
	04	Maarten van Gompel (RUN), Context as Linguistic Bridges
	05	Yulong Pei (TU/e), On local and global structure mining
	06	Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
	07	Wim van der Vegt (OU), Towards a software architecture for reusable game components

08 Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search

09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research

10 Alifah Syamsiyah (TU/e), In-database Preprocessing for Process Mining

11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models

12 Ward van Breda (VUA), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment

13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases

15 Konstantinos Georgiadis (OU), Smart CAT: Machine Learning for Configurable Assessments in Serious Games

16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling

17 Daniele Di Mitri (OU), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences

18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems

19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems

20 Albert Hankel (VUA), Embedding Green ICT Maturity in Organisations

21 Karine da Silva Miras de Araujo (VUA), Where is the robot?: Life as it could be

22 Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar

23 Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging

24 Lenin da Nóbrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots

25 Xin Du (TU/e), The Uncertainty in Exceptional Model Mining

26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization

	27	Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
	28	Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
	29	Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
	30	Bob Zadok Blok (UL), Creatief, Creatiever, Creatiefst
	31	Gongjin Lan (VUA), Learning better – From Baby to Better
	32	Jason Rhuggenaath (TU/e), Revenue management in online markets: pricing and online advertising
	33	Rick Gilsing (TU/e), Supporting service-dominant business model evaluation in the context of business model innovation
	34	Anna Bon (UM), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
	35	Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production
2021	01	Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space
	02	Rijk Mercuur (TUD), Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models
	03	Seyyed Hadi Hashemi (UvA), Modeling Users Interacting with Smart Devices
	04	Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning
	05	Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems
	06	Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot
	07	Armel Lefebvre (UU), Research data management for open science
	08	Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking
	09	Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play
	10	Quinten Meertens (UvA), Misclassification Bias in Statistical Learning
	11	Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision
	12	Lei Pi (UL), External Knowledge Absorption in Chinese SMEs

13 Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

14 Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support

15 Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm

16 Esam A. H. Ghaleb (UM), Bimodal emotion recognition from audio-visual cues

17 Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks

18 Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

19 Roberto Verdecchia (VUA), Architectural Technical Debt: Identification and Management

20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

21 Pedro Thiago Timbó Holanda (CWI), Progressive Indexes

22 Sihang Qiu (TUD), Conversational Crowdsourcing

23 Hugo Manuel Proença (UL), Robust rules for prediction and description

24 Kaijie Zhu (TU/e), On Efficient Temporal Subgraph Query Processing

25 Eoin Martino Grua (VUA), The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

26 Benno Kruit (CWI/VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables

27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You

28 Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

---

2022 01 Judith van Stegeren (UT), Flavor text generation for role-playing video games

02 Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey

03 Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare

04 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework

05 Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization

06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding

07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics

08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design

09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach

10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines

11 Mirjam de Haas (UT), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring

12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases

13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge

14 Michiel Overeem (UU), Evolution of Low-Code Platforms

15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining

16 Pieter Gijsbers (TU/e), Systems for AutoML Research

17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification

18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation

19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation

20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media

21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments

22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations

23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents

24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

25 Anna L.D. Latour (UL), Optimal decision-making under constraints and uncertainty

26 Anne Dirkson (UL), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences

27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences

28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems

29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality

30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays

31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

32 Cezara Pastrav (UU), Social simulation for socio-ecological systems

33 Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management

34 Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change

35 Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

---

2023 01 Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions

02 Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts

03 Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations

04 Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval

05 Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization, Including Real-World Medical Applications

06 António Pereira Barata (UL), Reliable and Fair Machine Learning for Risk Assessment

07 Tianjin Huang (TU/e), The Roles of Adversarial Examples on Trustworthiness of Deep Learning

08 Lu Yin (TU/e), Knowledge Elicitation using Psychometric Learning

09 Xu Wang (VUA), Scientific Dataset Recommendation with Semantic Techniques

10 Dennis J.N.J. Soemers (UM), Learning State-Action Features for General Game Playing

11 Fawad Taj (VUA), Towards Motivating Machines: Computational Modeling of the Mechanism of Actions for Effective Digital Health Behavior Change Applications

12 Tessel Bogaard (VUA), Using Metadata to Understand Search Behavior in Digital Libraries

13 Injy Sarhan (UU), Open Information Extraction for Knowledge Representation

14 Selma Čaušević (TUD), Energy resilience through self-organization

15 Alvaro Henrique Chaim Correia (TU/e), Insights on Learning Tractable Probabilistic Graphical Models

16 Peter Blomsma (TiU), Building Embodied Conversational Agents: Observations on human nonverbal behaviour as a resource for the development of artificial characters

17 Meike Nauta (UT), Explainable AI and Interpretable Computer Vision – From Oversight to Insight

18 Gustavo Penha (TUD), Designing and Diagnosing Models for Conversational Search and Recommendation

19 George Aalbers (TiU), Digital Traces of the Mind: Using Smartphones to Capture Signals of Well-Being in Individuals

20 Arkadiy Dushatskiy (TUD), Expensive Optimization with Model-Based Evolutionary Algorithms applied to Medical Image Segmentation using Deep Learning

21 Gerrit Jan de Bruin (UL), Network Analysis Methods for Smart Inspection in the Transport Domain

22 Alireza Shojaifar (UU), Volitional Cybersecurity

23 Theo Theunissen (UU), Documentation in Continuous Software Development

24 Agathe Balayn (TUD), Practices Towards Hazardous Failure Diagnosis in Machine Learning

25 Jurian Baas (UU), Entity Resolution on Historical Knowledge Graphs

26 Loek Tonnaer (TU/e), Linearly Symmetry-Based Disentangled Representations and their Out-of-Distribution Behaviour

27 Ghada Sokar (TU/e), Learning Continually Under Changing Data Distributions

28 Floris den Hengst (VUA), Learning to Behave: Reinforcement Learning in Human Contexts

29 Tim Draws (TUD), Understanding Viewpoint Biases in Web Search Results

---

2024 01 Daphne Miedema (TU/e), On Learning SQL: Disentangling concepts in data systems education

02 Emile van Krieken (VUA), Optimisation in Neurosymbolic Learning Systems

03 Feri Wijayanto (RUN), Automated Model Selection for Rasch and Mediation Analysis

04 Mike Huisman (UL), Understanding Deep Meta-Learning

05 Yiyong Gou (UM), Aerial Robotic Operations: Multi-environment Cooperative Inspection & Construction Crack Autonomous Repair

06 Azqa Nadeem (TUD), Understanding Adversary Behavior via XAI: Leveraging Sequence Clustering to Extract Threat Intelligence

07 Parisa Shayan (TiU), Modeling User Behavior in Learning Management Systems

08 Xin Zhou (UvA), From Empowering to Motivating: Enhancing Policy Enforcement through Process Design and Incentive Implementation

09 Giso Dal (UT), Probabilistic Inference Using Partitioned Bayesian Networks

10 Cristina-Iulia Bucur (VUA), Linkflows: Towards Genuine Semantic Publishing in Science

11 withdrawn

12 Peide Zhu (TUD), Towards Robust Automatic Question Generation For Learning

13 Enrico Liscio (TUD), Context-Specific Value Inference via Hybrid Intelligence

14 Larissa Capobianco Shimomura (TU/e), On Graph Generating Dependencies and their Applications in Data Profiling

15 Ting Liu (VUA), A Gut Feeling: Biomedical Knowledge Graphs for Interrelating the Gut Microbiome and Mental Health

16 Arthur Barbosa Câmara (TUD), Designing Search-as-Learning Systems

17 Razieh Alidoosti (VUA), Ethics-aware Software Architecture Design

18 Laurens Stoop (UU), Data Driven Understanding of Energy-Meteorological Variability and its Impact on Energy System Operations

19 Azadeh Mozafari Mehr (TU/e), Multi-perspective Conformance Checking: Identifying and Understanding Patterns of Anomalous Behavior

20 Ritsart Anne Plantenga (UL), Omgang met Regels

21 Federica Vinella (UU), Crowdsourcing User-Centered Teams

22 Zeynep Ozturk Yurt (TU/e), Beyond Routine: Extending BPM for Knowledge-Intensive Processes with Controllable Dynamic Contexts

23 Jie Luo (VUA), Lamarck's Revenge: Inheritance of Learned Traits Improves Robot Evolution

24 Nirmal Roy (TUD), Exploring the effects of interactive interfaces on user search behaviour

25 Alisa Rieger (TUD), Striving for Responsible Opinion Formation in Web Search on Debated Topics

26 Tim Gubner (CWI), Adaptively Generating Heterogeneous Execution Strategies using the VOILA Framework

27 Lincen Yang (UL), Information-theoretic Partition-based Models for Interpretable Machine Learning

28 Leon Helwerda (UL), Grip on Software: Understanding development progress of Scrum sprints and backlogs

29 David Wilson Romero Guzman (VUA), The Good, the Efficient and the Inductive Biases: Exploring Efficiency in Deep Learning Through the Use of Inductive Biases

30 Vijanti Ramautar (UU), Model-Driven Sustainability Accounting

31 Ziyu Li (TUD), On the Utility of Metadata to Optimize Machine Learning Workflows

32 Vinicius Stein Dani (UU), The Alpha and Omega of Process Mining

33 Siddharth Mehrotra (TUD), Designing for Appropriate Trust in Human-AI interaction

34 Robert Deckers (VUA), From Smallest Software Particle to System Specification - MuDForM: Multi-Domain Formalization Method

35 Sicui Zhang (TU/e), Methods of Detecting Clinical Deviations with Process Mining: a fuzzy set approach

36 Thomas Mulder (TU/e), Optimization of Recursive Queries on Graphs

37 James Graham Nevin (UvA), The Ramifications of Data Handling for Computational Models

38 Christos Koutras (TUD), Tabular Schema Matching for Modern Settings

39 Paola Lara Machado (TU/e), The Nexus between Business Models and Operating Models: From Conceptual Understanding to Actionable Guidance

40 Montijn van de Ven (TU/e), Guiding the Definition of Key Performance Indicators for Business Models

41 Georgios Siachamis (TUD), Adaptivity for Streaming Dataflow Engines

42 Emmeke Veltmeijer (VUA), Small Groups, Big Insights: Understanding the Crowd through Expressive Subgroup Analysis

43 Cedric Waterschoot (KNAW Meertens Instituut), The Constructive Conundrum: Computational Approaches to Facilitate Constructive Commenting on Online News Platforms

44 Marcel Schmitz (OU), Towards learning analytics-supported learning design

45 Sara Salimzadeh (TUD), Living in the Age of AI: Understanding Contextual Factors that Shape Human-AI Decision-Making

46 Georgios Stathis (Leiden University), Preventing Disputes: Preventive Logic, Law & Technology

47 Daniel Daza (VUA), Exploiting Subgraphs and Attributes for Representation Learning on Knowledge Graphs

48 Ioannis Petros Samiotis (TUD), Crowd-Assisted Annotation of Classical Music Compositions

---

2025 01 Max van Haastrecht (UL), Transdisciplinary Perspectives on Validity: Bridging the Gap Between Design and Implementation for Technology-Enhanced Learning Systems

02 Jurgen van den Hoogen (JADS), Time Series Analysis Using Convolutional Neural Networks

03 Andra-Denis Ionescu (TUD), Feature Discovery for Data-Centric AI

04 Rianne Schouten (TU/e), Exceptional Model Mining for Hierarchical Data

05 Nele Albers (TUD), Psychology-Informed Reinforcement Learning for Situated Virtual Coaching in Smoking Cessation

06 Daniël Vos (TUD), Decision Tree Learning: Algorithms for Robust Prediction and Policy Optimization

07 Ricky Maulana Fajri (TU/e), Towards Safer Active Learning: Dealing with Unwanted Biases, Graph-Structured Data, Adversary, and Data Imbalance

08 Stefan Bloemheuvel (TiU), Spatio-Temporal Analysis Through Graphs: Predictive Modeling and Graph Construction

09 Fadime Kaya (VUA), Decentralized Governance Design - A Model-Based Approach

10 Zhao Yang (UL), Enhancing Autonomy and Efficiency in Goal-Conditioned Reinforcement Learning

11 Shahin Sharifi Noorian (TUD), From Recognition to Understanding: Enriching Visual Models Through Multi-Modal Semantic Integration

12 Lijun Lyu (TUD), Interpretability in Neural Information Retrieval

13 Fuda van Diggelen (VUA), Robots Need Some Education: on the complexity of learning in evolutionary robotics

14 Gennaro Gala (TU/e), Probabilistic Generative Modeling with Latent Variable Hierarchies

15 Michiel van der Meer (UL), Opinion Diversity through Hybrid Intelligence

16 Monika Grewal (TU Delft), Deep Learning for Landmark Detection, Segmentation, and Multi-Objective Deformable Registration in Medical Imaging

17 Matteo De Carlo (VUA), Real Robot Reproduction: Towards Evolving Robotic Ecosystems

18 Anouk Neerincx (UU), Robots That Care: How Social Robots Can Boost Children's Mental Wellbeing

19 Fang Hou (UU), Trust in Software Ecosystems

20 Alexander Melchior (UU), Modelling for Policy is More Than Policy Modelling (The Useful Application of Agent-Based Modelling in Complex Policy Processes)

21 Mandani Ntekouli (UM), Bridging Individual and Group Perspectives in Psychopathology: Computational Modeling Approaches using Ecological Momentary Assessment Data

22 Hilde Weerts (TU/e), Decoding Algorithmic Fairness: Towards Interdisciplinary Understanding of Fairness and Discrimination in Algorithmic Decision-Making

23 Roderick van der Weerdt (VUA), IoT Measurement Knowledge Graphs: Constructing, Working and Learning with IoT Measurement Data as a Knowledge Graph

24 Zhong Li (UL), Trustworthy Anomaly Detection for Smart Manufacturing

25 Kyana van Eijndhoven (TiU), A Breakdown of Breakdowns: Multi-Level Team Coordination Dynamics under Stressful Conditions

26 Tom Pepels (UM), Monte-Carlo Tree Search is Work in Progress

27 Danil Provodin (JADS, TU/e), Sequential Decision Making Under Complex Feedback

28 Jinke He (TU Delft), Exploring Learned Abstract Models for Efficient Planning and Learning

29 Erik van Haeringen (VUA), Mixed Feelings: Simulating Emotion Contagion in Groups

30 Myrthe Reuver (VUA), A Puzzle of Perspectives: Interdisciplinary Language Technology for Responsible News Recommendation

31 Gebrekirstos Gebreselassie Gebremeskel (RUN), Spotlight on Recommender Systems: Contributions to Selected Components in the Recommendation Pipeline

32 Ryan Brate (UU), Words Matter: A Computational Toolkit for Charged Terms

33 Merle Reimann (VUA), Speaking the Same Language: Spoken Capability Communication in Human-Agent and Human-Robot Interaction

34 Eduard C. Groen (UU), Crowd-Based Requirements Engineering

35 Urja Khurana (VUA), From Concept To Impact: Toward More Robust Language Model Deployment

36 Anna Maria Wegmann (UU), Say the Same but Differently: Computational Approaches to Stylistic Variation and Paraphrasing

37 Chris Kamphuis (RUN), Exploring Relations and Graphs for Information Retrieval

38 Valentina Maccatrozzo (VUA), Break the Bubble: Semantic Patterns for Serendipity

39 Dimitrios Alivanistos (VUA), Knowledge Graphs & Transformers for Hypothesis Generation: Accelerating Scientific Discovery in the Era of Artificial Intelligence

40 Stefan Grafberger (UvA), Declarative Machine Learning Pipeline Management via Logical Query Plans

41 Mozhgan Vazifehdoostirani (TU/e), Leveraging Process Flexibility to Improve Process Outcome - From Descriptive Analytics to Actionable Insights

42 Margherita Martorana (VUA), Semantic Interpretation of Dataless Tables: a metadata-driven approach for findable, accessible, interoperable and reusable restricted access data

43 Krist Shingjergji (OU), Sense the Classroom - Using AI to Detect and Respond to Learning-Centered Affective States in Online Education

44 Robbert Reijnen (TU/e), Dynamic Algorithm Configuration for Machine Scheduling Using Deep Reinforcement Learning

45 Anjana Mohandas Sheeladevi (VUA), Occupant-Centric Energy Management: Balancing Privacy, Well-being and Sustainability in Smart Buildings

46 Ya Song (TU/e), Graph Neural Networks for Modeling Temporal and Spatial Dimensions in Industrial Decision-making

47 Tom Kouwenhoven (UL), Collaborative Meaning-Making. The Emergence of Novel Languages in Humans, Machines, and Human-Machine Interactions

48 Evy van Weelden (TiU), Integrating Virtual Reality and Neurophysiology in Flight Training

49 Selene Báez Santamaría (VUA), Knowledge-centered conversational agents with a drive to learn

50 Lea Krause (VUA), Contextualising Conversational AI

51 Jiaxu Zhao (TU/e), Understanding and Mitigating Unwanted Biases in Generative Language Models

52 Qiao Xiao (TU/e), Model, Data and Communication Sparsity for Efficient Training of Neural Networks

53 Gaole He (TUD), Towards Effective Human-AI Collaboration: Promoting Appropriate Reliance on AI Systems

54 Go Sugimoto (VUA), MISSING LINKS Investigating the Quality of Linked Data and its Tools in Cultural Heritage and Digital Humanities

55 Sietze Kai Kuilman (TUD), AI that Glitters is Not Gold: Requirements for Meaningful Control of AI Systems

56 Wijnand van Woerkom (UU), A Fortiori Case-Based Reasoning: Formal Studies with Applications in Artificial Intelligence and Law

57 Syeda Amna Sohail (UT), Privacy-Utility Trade-Off in Healthcare Metadata Sharing and Beyond: A Normative and Empirical Evaluation at Inter and Intra Organizational Levels

58 Junhan Wen (TUD), "From iMage to Market": Machine-Learning-Empowered Fruit Supply

59 Mohsen Abbaspour Onari (TU/e), From Explanation to Trust: Modeling and Measuring Trust in Explainable Decision Support

60 Marcel Jurriaan Robeer (UU), Beyond Trust: A Causal Approach to Explainable AI in Law Enforcement

61 Shuai Wang (VUA), Links in Large Integrated Knowledge Graphs: Analysis, Refinement, and Domain Applications

62 Khaleel Asyraaf Mat Sanusi (OU), Augmenting a learning model within immersive learning environments for psychomotor skills

63 Rashid Zaman (TU/e), Online Conformance Checking on Degraded Data

64 Jens d'Hondt (TU/e), Effective and Efficient Multivariate Similarity Search

65 Aswin Balasubramaniam (UT), Disentangling Runner Drone Interaction Potentialities

---

2026 01 Pei-Yu Chen (TUD), Human-Agent Alignment Dialogues: Eliciting User Information at Runtime for Personalized Behavior Support

02 Hezha Hassan Mohammedkhan (TiU), Estimating Body Measurements of Children from 2D Images: Towards the Automatic Detection of Malnutrition

03 Kyriakos Psarakis (TUD), Democratizing Scalable Cloud Applications: Transactional Stateful Functions on Streaming Dataflows

04 Boyu Xu (UU), Exploring Indirect Relations Between Topics in Neuroscience Literature Using Augmented Reality to Inform Experimental Design

05 Koen Minartz (TU/e), Stochastic Simulation with Geometric Deep Generative Models

06 Azim Afroozeh (CWI, VUA), FastLanes: A Next-Gen File Format

07 Inès Blin (VUA), Narrative Understanding with Knowledge Graphs

08 Paul van Vulpen (UU), Debating Digital Dominance: Decentralized Technology Governance For Strategic Autonomy

09 Afrizal Doewes (TU/e), Rethinking Automated Essay Scoring: Agreement, Fairness, and Feedback

10 Nikolaos Delapaschos Kondylidis (VUA), Establishing Task-Oriented Understanding between Agents

11 Işıl Baysal Erez (UT), Handling Missing Data with Meta-Learning and Large Language Models

12 Xue Li (UvA), From Fine-tuning to Prompting: A Paradigm Shift in Knowledge Graph Construction

13 Isaac da Silva Torres (VUA), Guidelines To Flux Between Conceptual Models: Understanding Complex Digital Business Ecosystems

14 Philip Lippmann (TUD), Synthetic Data for Robust Language Modelling

15 Rashmi Khazanchi (OU), Artificial Intelligence in Education: Impact of AI-Based Systems on Mathematics Achievement

SUMMARY

Knowledge graphs (KGs) have become a central technology for integrating, structuring, and reasoning over information. Yet, constructing KGs in complex domains remains a significant challenge. These domains, such as organizational conversations, are characterized by noisy, evolving, and structurally diverse data, which makes traditional workflows brittle and resource-intensive. This thesis investigates how knowledge graph construction (KGC) can be made more robust and efficient by examining the limitations of the pretrain-then-finetune (PTFT) paradigm and by designing solutions based on the emerging pretrain, prompt, and predict (PPP) paradigm. The first part of the thesis focuses on the challenges of PTFT-based task-specific models. Chapter 2 studies distribution shifts in Named Entity Recognition (NER) and demonstrates how even modest changes in data distributions can lead to large performance drops, while also showing that shift measures can predict such failures. Chapter 3 evaluates static topic models for detecting the emergence of new topics in dynamic corpora, finding that while neural models capture semantic coherence well, they struggle to identify when new topics arise. Chapter 4 investigates cross-document coreference in multi-party email, surfacing difficulties caused by sparse mentions, aliases, and long conversational structure, and showing that PTFT models remain inadequate even with fine-tuning. Together, these chapters reveal why task-specific models generalize poorly in real-world, complex settings. The second part of the thesis turns to PPP-based solutions that leverage large language models (LLMs) through prompting. Chapter 5 examines instruction-tuned LLMs for relation extraction, showing that they achieve competitive results when schema knowledge is made explicit in prompts, while also highlighting the limitations of traditional evaluation metrics in this setting. Chapter 6 introduces knowledge-centric prompt composition for knowledge base construction, demonstrating how prompts enriched with schema constraints and examples improve precision without the need for fine-tuning. Chapter 7 presents TableSwift, a hybrid system for data preparation that routes tasks between LLM-generated code and lightweight fallbacks, reducing cost while maintaining accuracy across tasks such as transformation, error detection, and entity matching. These chapters illustrate how prompt-based workflows can provide flexible, adaptable, and cost-aware alternatives to PTFT pipelines. This thesis thus traces a critical paradigm shift in KGC: from specialized PTFT pipelines toward PPP workflows that are more flexible and interpretable. By diagnosing the weaknesses of PTFT models in conversational data and proposing PPP-based orchestration methods for extraction and data preparation, it contributes both empirical insights and practical tools for build-

ing reliable knowledge graphs. While challenges remain, such as integrating multimodal evidence, handling dynamic updates, and developing more robust evaluation protocols, the work presented here demonstrates concrete steps toward making KGC more resilient, efficient, and suited to the complexities of real-world data.

SAMENVATTING

Kennisgrafen (KG's) zijn een centrale technologie geworden voor het integreren, structureren en redeneren over informatie. Het construeren van kennisgrafen in complexe domeinen blijft echter een grote uitdaging. Deze domeinen, zoals organisatorische conversaties, worden gekenmerkt door rommelige, evoluerende en structureel diverse data, waardoor traditionele workflows kwetsbaar en kostbaar zijn. Dit proefschrift onderzoekt hoe de constructie van kennisgrafen (Knowledge Graph Construction, KGC) robuuster en efficiënter kan worden gemaakt door zowel de beperkingen van het pretrain-then-finetune (PTFT) paradigma te analyseren als oplossingen te ontwerpen op basis van het opkomende pretrain, prompt, and predict (PPP) paradigma. Het eerste deel van dit proefschrift richt zich op de uitdagingen van PTFT-gebaseerde, taakspecifieke modellen. Hoofdstuk 2 bestudeert distributieverschuivingen in Named Entity Recognition (NER) en laat zien dat zelfs kleine verschuivingen in data tot grote prestatieverliezen kunnen leiden, terwijl ook wordt aangetoond dat shiftmaten dergelijke fouten kunnen voorspellen. Hoofdstuk 3 evalueert statische topicmodellen voor het detecteren van het ontstaan van nieuwe onderwerpen in dynamische corpora. Hieruit blijkt dat neurale modellen weliswaar semantische samenhang goed vastleggen, maar moeite hebben om te bepalen wanneer nieuwe onderwerpen opkomen. Hoofdstuk 4 onderzoekt cross-document coreference in e-mailconversaties met meerdere deelnemers en brengt moeilijkheden aan het licht, zoals schaarse verwijzingen, aliassen en lange gesprekstructuren. Dit toont aan dat PTFT-modellen zelfs na fine-tuning onvoldoende presteren. Gezamenlijk laten deze hoofdstukken zien waarom taakspecifieke modellen slecht generaliseren in realistische, complexe omgevingen. Het tweede deel van het proefschrift richt zich op PPP-gebaseerde oplossingen die gebruikmaken van large language models (LLM's) via prompting. Hoofdstuk 5 onderzoekt instruction-tuned LLM's voor relatie-extractie en laat zien dat ze competitieve resultaten behalen wanneer schema-kennis expliciet wordt opgenomen in prompts, maar benadrukt ook de beperkingen van traditionele evaluatiemethoden in deze context. Hoofdstuk 6 introduceert knowledge-centric prompt composition voor knowledge base construction, waarbij prompts verrijkt met schema-constraints en voorbeelden leiden tot betere precisie zonder dat fine-tuning nodig is. Hoofdstuk 7 presenteert TableSwift, een hybride systeem voor datavoorbewerking dat taken routet tussen door LLM's gegenereerde code en lichtere fallback-methoden. Dit systeem verlaagt de kosten aanzienlijk, terwijl het toch een hoge nauwkeurigheid behoudt voor taken zoals transformatie, foutdetectie en entiteitsovereenkomst. Deze hoofdstukken illustreren hoe prompt-gebaseerde workflows flexibele, aanpasbare en kostenefficiënte alterna-

tieven kunnen bieden voor PTFT-pijplijnen. Dit proefschrift volgt daarmee een cruciale paradigmaverschuiving in KGC: van gespecialiseerde PTFT-pijplijnen naar PPP-workflows die flexibeler en beter uitlegbaar zijn. Door de zwaktes van PTFT-modellen in conversatiedata bloot te leggen en PPP-gebaseerde orkestratiemethoden te ontwikkelen voor extractie en datavoorbewerking, levert dit werk zowel empirische inzichten als praktische hulpmiddelen voor het bouwen van betrouwbare kennisgrafen. Hoewel er nog uitdagingen blijven, zoals het integreren van multimodale data, het omgaan met dynamische updates en het ontwikkelen van robuustere evaluatieprotocollen, laat dit proefschrift concrete stappen zien om KGC veerkrachtiger, efficiënter en beter geschikt te maken voor de complexiteit van data uit de echte wereld.