High-coverage Information Extraction from Web and Narrative Texts

A dissertation submitted towards the degree
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by

Sneha Singhania

Saarbrücken, 2025

Declaration of original authorship

I hereby declare that this dissertation is my own original work except where otherwise indicated. All data or concepts drawn directly or indirectly from other sources have been correctly acknowledged. This dissertation has not been submitted in its present or similar form to any other academic institution either in Germany or abroad for the award of any other degree.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, December 2025

gez. Sneha Singhania

Defense Colloqium

Date:	24.02.2026
Dean of the Faculty:	Univ.-Professor Dr. Roland Speicher

Examination Committee

Chair of the Committee:	Prof. Dr. Sven Apel
Reviewers:	Prof. Dr. Martin Theobald
	Prof. Dr. Simon Razniewski
	Prof. Dr. Gerhard Weikum
Academic Assistant:	Dr. Frances Yung

Abstract

The web predominantly stores information in unstructured forms, supporting applications such as search, large language model (LLM) training, and decision-making tools. Information extraction (IE) aims to transform key textual content into structured representations, such as subject-predicate-object triples. Most IE systems have largely prioritized precision, often at the expense of recall. As LLMs and knowledge-intensive applications become more prominent, there is a growing need for frameworks that achieve high recall—capturing all relevant facts across diverse sources, formats, and contexts. This dissertation advances methods for high-recall IE across different settings: web-scale documents, parametric LLM knowledge, and long-form narrative texts.

To address web-scale extraction, we introduce a new task: *predicting the information coverage of a document* for relation extraction. We propose Herb, a lightweight classifier that identifies high-recall documents using a combination of explicit features and latent document-level signals. Rather than performing exhaustive extraction, Herb prioritizes documents by estimated coverage, maximizing information yield under strict budget constraints. We also investigate IE from temporally evolving web-scale documents, focusing on news articles. We present Neon, a framework that extracts OpenIE-style propositions to construct an entity-centric, timestamped representation. Integrating these propositions into a retrieval-augmented generation (RAG) framework improves question answering over dynamic information.

To harness parametric knowledge in LLMs, we study high-recall knowledge extraction by probing LLMs and solve the multi-valued slot-filling task, where a subject–relation pair can have multiple correct objects. While prior methods have focused on extracting a single object, we formulate the problem as a *rank-then-select* task. We develop predicate-specific prompting techniques that improve the extraction of valid objects for multi-valued relations.

For long-form narratives, where evidence is sparse and dispersed, we address the challenge of *extracting long lists of objects*. Our proposed L3X framework combines recall-oriented generation—via RAG, iterative prompting, and pseudo-relevance feedback—with a precision-oriented scrutinization stage. This architecture yields substantially higher recall while maintaining good precision.

By rethinking the interplay between retrieval and extraction, this dissertation advances the state-of-the-art in high-recall IE. The core contributions include novel extraction methods, large-scale task-specific benchmarks, empirical results that push the boundary of extraction capabilities, and demonstration systems that reveal persistent challenges and opportunities for the next generation of high-recall IE.

Kurzfassung

Das Web speichert Informationen vorwiegend in unstrukturierter Form und unterstützt Anwendungen wie die Suche, das Training großer Sprachmodelle (Large Language Models: LLMs) und Entscheidungsfindung. Informationsextraktion (IE) zielt darauf ab, zentrale textuelle Inhalte in strukturierte Repräsentationen, wie etwa Subjekt-Prädikat-Objekt-Tripel, zu transformieren. Bislang priorisierten die meisten IE-Systeme die Metrik Precision, oft auf Kosten des Recalls. Da LLMs und wissensintensive Anwendungen zunehmend an Bedeutung gewinnen, wächst der Bedarf an Frameworks, die einen hohen Recall erzielen—also das Erfassen aller relevanten Fakten über diverse Quellen, Formate und Kontexte hinweg. Diese Dissertation entwickelt Methoden für High-Recall-IE in verschiedenen Szenarien weiter: große Mengen an Webseiten, parametrisches Wissen in LLMs und narrative Langtexte.

Um die Extraktion aus Web-Dokumenten zu adressieren, führen wir eine neue Aufgabe ein: die *Vorhersage der Informationsabdeckung eines Dokuments* für die Relationsextraktion. Wir stellen Herb vor, einen leichtgewichtigen Klassifikator, der High-Recall-Dokumente mittels einer Kombination aus expliziten Merkmalen und latenten Signalen auf Dokumentenebene identifiziert. Anstatt eine erschöpfende Extraktion durchzuführen, priorisiert Herb Dokumente anhand ihrer geschätzten Abdeckung und maximiert so die Informationsausbeute unter strikten Budgetbeschränkungen. Zudem untersuchen wir IE in sich temporal verändernden Webinhalten, mit Fokus auf Nachrichtenartikeln. Wir präsentieren Neon, ein Framework, das Propositionen extrahiert, um eine entitätszentrierte, temporal angereicherte Repräsentation zu konstruieren. Die Integration dieser Propositionen in ein Framework für Retrieval-Augmented Generation (RAG) verbessert die Beantwortung von Fragen über dynamische Informationen.

Um parametrisches Wissen in LLMs zu erschließen, untersuchen wir High-Recall-IE mittels LLM-Probing. Wir adressieren dabei die Aufgabe des Multi-Valued Slot-Filling, bei dem ein Subjekt-Relation-Paar mehrere korrekte Objekte aufweisen kann. Während bisherige Methoden typischerweise nur ein einzelnes Objekt extrahieren, formulieren wir diese Aufgabe als *Rank-then-Select*-Problem. Wir entwickeln prädikatsspezifische Prompting-Techniken, die die Extraktion valider Objekte für mehrwertige Relationen verbessern.

Bei narrativen Langtexten, in denen Evidenz spärlich und verstreut ist, adressieren wir die Herausforderung, *lange Listen von Objekten zu extrahieren*. Unser L3X-Framework kombiniert Recall-orientierte Generierung—mittels RAG, iterativem Prompting und Pseudo-Relevance Feedback—mit einer Precision-orientierten Überprüfungsphase. Diese Architektur liefert einen wesentlich höheren Recall bei gleichzeitiger Wahrung guter Precision.

In all diesen Bereichen des Zusammenspiels von Suche und Extraktion bringt diese Dissertation den State-of-the-Art in High-Recall-IE voran. Die Kernbeiträge umfassen neuartige Extraktionsmethoden, groß angelegte aufgabenspezifische Benchmarks, empirische Ergebnisse, die die Grenzen der Extraktionsfähigkeiten erweitern, sowie Demonstrationssysteme, die bestehende Herausforderungen und Chancen für die nächste Generation in High-Recall-IE aufzeigen.

Acknowledgements

I am extremely grateful to my advisors, Simon and Gerhard, for their invaluable mentorship. Starting my PhD at the height of the pandemic was challenging, and I am especially thankful to Simon for his persistent encouragement and support. You taught me how to break down complex problems into simple and intuitive ideas. The freedom you gave me allowed me to explore my interests and carve my own career path. Thank you, Gerhard, for your exceptional technical guidance, thought-provoking discussions, and for always being available. You taught me to stay grounded and to critically evaluate research trends, helping me develop my own perspectives. I am truly grateful to both of you for shaping me as a researcher and making this journey so rewarding.

I sincerely thank Martin for serving on my doctoral committee and reviewing my thesis.

It was a lot of fun to pursue two internships during my PhD. First at Apple, and then at Microsoft Research. These experiences gave me firsthand exposure to applied research and the chance to interact with leading researchers in the field. I am thankful to my mentors at both places for their guidance and support.

A heartfelt thanks to my D5 friends and colleagues. Our passionate kicker playoffs, reading group sessions, and lunchtime banter made this journey all the more enjoyable. I learned so much from each of you and truly appreciate your support.

To my incredible friends, my "Barricaded Menschen" crew, Shrestha, Arpan, Florena, thank you for making life in Saarbrücken and our European adventures unforgettable. Raouf, thank you for always having my back and for our endless conversations about life and beyond. Abhinav, from our IIIT-B days to California, thank you for being a constant source of laughter and support.

And last but not least, I'm deeply indebted to my family. Thanks so much mom and dad, Seema and Ajay, for your unconditional love, patience, and immense faith in my abilities. Vasu and Savi, thank you for always putting up with my shenanigans and for supporting me in your own unique ways. Vishal Mamaji, I am grateful for your support since always. And finally, thank you, Apoorv, for being my greatest champion and pillar of support.

Contents

1 Introduction

1.1 Motivation

The Web contains information in both unstructured and structured formats. Structured knowledge is often preferred by humans, organizations, and even artificial intelligence (AI) systems since it is more understandable, logically organizable, and efficiently retrievable and searchable. However, the vast majority of information on the web remains unstructured, and decades of research have focused on converting unstructured content into structured form (Han et al., 2020). One way to achieve this is through *information extraction* (IE) frameworks, which automatically transforms unstructured text into structured knowledge in the form of (`subject`, `predicate`, `object`) triples, SPO triples for short. These structured triples are typically stored in knowledge graphs (KGs), or knowledge bases (KBs) (Weikum et al., 2021; Hogan et al., 2021).

A perfect IE system would extract *all* factually correct triples. For instance, consider the first few passages from the Wikipedia article on *APJ Abdul Kalam*[1], given in Figure 1.1. For the subject *APJ Abdul Kalam*, Table 1.1 gives all the SPO triples extractable from the passage. Having these facts stored in a KB enables search, question answering and analytics-based decision support. Yet, existing large-scale open-sourced KBs such as Wikidata (Vrandečić and Krötzsch, 2014), remain highly incomplete (Galárraga et al., 2017). The Wikidata entry for *APJ Abdul Kalam*[2] does not contain any information on his organizational memberships, while the Wikipedia article mentions multiple of them (also shown in Table 1.1). This is partly because KBs require near-perfect precision for downstream usage. As a result, they are either manually curated or populated using IE systems optimized for high confidence, which often leads to *low recall*. Recall is defined as the fraction of correct facts extracted compared to total ground truth SPO triples.

With the advent of large language models (LLMs), including masked models like BERT (Devlin et al., 2019) and autoregressive (causal) models like GPT (Brown et al., 2020), the whole landscape of how people use and interact with web content has changed. Since these LLMs are pretrained on vast amounts of web data, one can directly prompt an LLM in natural language for downstream usage and retrieval, rather can explicitly relying on KBs. Work by Petroni et al. (2019), and follow-up works (Liu et al., 2023a), show how to probe LLMs for factual knowledge using cloze-style prompts. These methods again, however, optimize for precision@1 or hits@1 and completely ignore recall.

High recall, however, is crucial for trustworthy systems. Consider the prompt "Germany shares a border with `[MASK]`.", for which a reliable model should generate all neighboring countries, i.e.,

[1] https://en.wikipedia.org/wiki/A._P._J._Abdul_Kalam
[2] https://www.wikidata.org/wiki/Q9513

"Avul Pakir Jainulabdeen Abdul Kalam (15 October 1931 – 27 July 2015) was an Indian aerospace scientist and statesman who served as the president of India from 2002 to 2007.
Born and raised in a Muslim family in Rameswaram, Tamil Nadu, Kalam studied physics and aerospace engineering. He spent the next four decades as a scientist and science administrator, mainly at the Defence Research and Development Organisation (DRDO) and Indian Space Research Organisation (ISRO) and was intimately involved in India's civilian space programme and military missile development efforts. He was known as the "Missile Man of India" for his work on the development of ballistic missile and launch vehicle technology. He also played a pivotal organisational, technical, and political role in Pokhran-II nuclear tests in 1998, India's second such test after the first test in 1974."

Figure 1.1: Unstructured text snippet from APJ Abdul Kalam's Wikipedia article.

Subject	**Predicate**	**Object**
APJ Abdul Kalam	`aliasName`	Avul Pakir Jainulabdeen Abdul Kalam
APJ Abdul Kalam	`aliasName`	Kalam
APJ Abdul Kalam	`aliasName`	Missile Man of India
APJ Abdul Kalam	`bornIn`	Rameswaram
APJ Abdul Kalam	`bornIn`	Tamil Nadu
APJ Abdul Kalam	`positionHeld`	President of India
APJ Abdul Kalam	`fieldOfWork`	physics
APJ Abdul Kalam	`fieldOfWork`	aerospace engineering
APJ Abdul Kalam	`occupation`	aerospace scientist
APJ Abdul Kalam	`occupation`	statesman
APJ Abdul Kalam	`occupation`	science administrator
APJ Abdul Kalam	`partOf`	Defence Research & Development Organisation
APJ Abdul Kalam	`partOf`	Indian Space Research Organisation
APJ Abdul Kalam	`partOf`	Pokhran-II

Table 1.1: All SPO triples for the text in Figure 1.1 with APJ Abdul Kalam as the subject.

Denmark, Poland, Czech Republic, Austria, Switzerland, France, Luxembourg, Belgium, and Netherlands, with equal confidence to express that they are all equally true. However, this does not occur, as seen from prompting the BERT model[3]. The top-5 entities, along with their likelihood, predicted in the `[MASK]` position are: *"france (0.140), austria (0.072), poland (0.069), canada (0.060), switzerland (0.058)"*. Here, except the fourth entity, all are correct; however there is no easy way to select-and-prune correct entities from the token distribution. This occurs as LLM token probabilities are skewed by the frequency of entity co-occurrences in their training data, and a slight variation in prompts leads to inconsistent generations, making this whole task very challenging.

[3] https://huggingface.co/google-bert/bert-base-uncased?text=Germany+shares+a+border+with+%5BMASK%5D.

User:

Give me the list of friends of Harry Potter from all of Harry Potter book series. Generate only the names without additional details.

Assistant:

"Here is a list of Harry Potter's friends from the book series: [Ron Weasley, Hermione Granger, Neville Longbottom, Luna Lovegood, Ginny Weasley, Fred Weasley, George Weasley, Hagrid, Sirius Black, Remus Lupin, Cedric Diggory, Dean Thomas, Seamus Finnigan, Lee Jordan, Cho Chang, Dobby, Oliver Wood, Angelina Johnson, Alicia Spinnet, Katie Bell, Colin Creevey, Dennis Creevey, Aberforth Dumbledore, Susan Bones, Justin Finch-Fletchley, Ernie Macmillan, Hannah Abbott, Padma Patil, Parvati Patil, Terry Boot, Michael Corner, Zacharias Smith]"

Figure 1.2: Object generation using GPT-4o with (Harry Potter, `friend`) as the SP pair.

These problems intensify when performing IE, especially with LLMs or other machine learning models, over long documents (Liu et al., 2024; Xu et al., 2024b), domain-specific content such as book narratives (Chang et al., 2023), or when handling long-tail entities (Mallen et al., 2023). For example, even the most advanced LLM currently available (as of July 2025), GPT-4o, generates an incomplete list of 30 names for the query *"all friends of Harry Potter"*, and even includes an incorrect entity, as illustrated in Figure 1.2. In reality, there are more than 50 entities who can be associated as friends of *Harry Potter* in the eponymous book series. For instance, characters like "Ollivander" are clearly depicted as his supporters, as evidenced by the following passage: *"**Mr. Ollivander**, I'm sorry to disturb you," **Harry** said. "My dear boy." Ollivander's voice was feeble. "You rescued us. I thought we would die in that place. I can never thank you . . . never thank you . . . enough." "We were glad to do it." Harry's scar throbbed."* LLMs fail to generate such entities without sufficient supporting context: these characters occur sparsely across the books and are absent in the model's parametric knowledge. Explicitly adding relevant context in the prompt and performing retrieval-augmented generation (Guu et al., 2020; Lewis et al., 2020b) helps LLMs to tackle some of these issues (Li et al., 2024), but current methods are far from perfect and struggle with overall fact coverage.

This dissertation aims to develop recall-oriented IE systems that perform extraction across diverse domains, including web-scale corpora, LLM outputs, and long narrative texts, with the ultimate goal of quantifying and enhancing information coverage and recall.

1.2 Prior Work and Its Limitations

Estimating Recall. Estimating information coverage of any given text (be it from web content or domain-specific corpora) remains inherently difficult: it requires an exhaustive comparison against the underlying corpus as a gold standard. The limited prior work in this space has explored automatic identification of incomplete information using aggregate-level statistics (Razniewski et al., 2017; Galárraga et al., 2017), computing relative completeness across semantically similar entities (Balaraman et al., 2018), using textual features to assess whether a sentence or paragraph contains all objects for a given subject-predicate pair (Razniewski et al., 2019), and relying on explicit count quantifiers in the text (Mirza et al., 2018).

However, these methods fall short in accurately estimating the coverage of arbitrary text spans, especially when length and content significantly vary. It is unclear how these standalone methods would perform when operating under resource constraints. Since they rely on weak neural baselines, they can be brittle when facing temporal facts or inconsistencies across heterogeneous sources.

IE from Web. Relation extraction (RE) extracts subject-predicate-object (SPO) triples from unstructured text, where S and O are named entities (given), and P is a predicate connecting them (to be identified by the system). RE has been widely studied (Suchanek et al., 2009; Mintz et al., 2009; Riedel et al., 2010). Most state-of-the-art RE methods use neural models optimized for precision with limited recall (Han et al., 2020). These methods typically operate at the sentence level and are evaluated on benchmarks such as `SemEval` (Hendrickx et al., 2010), a small dataset with 10.7k examples and 9 relations, and `TACRED` (Zhang et al., 2017), a larger dataset with 106k samples and 41 relations. However, these benchmarks have several limitations: (i) designed for single-valued slot-filling task with annotated SO pairs; (ii) suffer from limited context; (iii) fail to handle long-tail entities; and (iv) do not capture multi-valued relations. Although methods competing on these benchmarks report precision, recall, and F1 scores (the harmonic mean of precision and recall), most are tuned for precision, thereby neglecting recall.

Open information extraction (OpenIE) has been proposed as a way to improve recall in automatic KB construction (Pei et al., 2023). Yet even the best-performing OpenIE systems (Manning et al., 2014; Kolluru et al., 2020a) struggle with relations that rely on sparse cues or long-range dependencies. Like standard RE, OpenIE methods also operate at the sentence level.

IE with LLMs. With advances in language modeling architectures, LLMs have been explored as implicit KBs (Petroni et al., 2019; Veseli et al., 2023a). Follow-up work has shown that LLMs fail to capture long-tail facts (Kandpal et al., 2023; Sun et al., 2024). Moreover, they exhibit similar recall deficiencies and disregard multi-valued relations. Early approaches treat LLMs as classifiers or sequence taggers over single passages, using cloze-style prompts to probe masked language models (MLMs). With MLMs, the generation is restricted to single-word or single-token responses.

More recent generative models can produce multi-word or multi-token outputs, but still struggle with long-context extraction—especially when the predicate cues are dispersed across multiple sentences or paragraphs. Furthermore, LLM outputs are not calibrated in terms of likelihood versus correctness (Jiang et al., 2021a), and different prompt templates often yield inconsistent generations (Elazar et al., 2021).

IE from Long Documents. Recent works (Zhao et al., 2024b; Xu et al., 2024a) extend the scope of extraction to larger input contexts under the theme of "long-distance IE", moving beyond sentence- or paragraph-level inputs. However, techniques like graph neural networks or LLM-based generative IE still target news articles or encyclopedic text, and remain ill-equipped for book-length narrative content. Even prominent document-level RE benchmarks, such as the human-annotated DocRED (Yao et al., 2019) and the automatically constructed REBEL (Huguet Cabot and Navigli, 2021), restrict inputs to single Wikipedia paragraphs.

Long documents can be processed either via retrieval-augmented generation (RAG) (Guu et al., 2020; Lewis et al., 2020b) or long-context (LC) models (OpenAI, 2023b; Reid et al., 2024). LC models can handle up to millions of tokens in a single LLM call. Both approaches present trade-offs (Li et al., 2024): (i) RAG incurs more LLM calls since contexts are chunked into contiguous passages; (ii) LC generation offers minimal traceability, making it difficult to interpret or explain which context segments influence predictions; (iii) RAG is more energy-efficient, whereas training and running LC models at inference leads to higher API costs. Moreover, the relative performance of RAG and LC depends on the downstream task (Xu et al., 2024b; Li et al., 2025b). Notably, no prior work specifically compares their performance for IE.

LLMs for IE over fiction—especially for character-centric information—have been studied (Bamman et al., 2019; Stammbach et al., 2022; Chang et al., 2023). But these works focus on named entity recognition (NER)-style generation of single names from isolated passages. Broader cultural-analytic applications using LLMs have been explored (Piper and Bagga, 2024; Bamman et al., 2024). However, none of these methods tackle full-fledged RE over book-length texts.

1.3 Challenges

Building IE systems that are highly accurate and can extract complete sets requires navigating inherent trade-offs and uncertainties. First, there is the fundamental *precision-recall tradeoff*: as systems become more inclusive, by lowering model's decision threshold to maximize recall, the risk of introducing false positives inevitably increases, leading to lower precision. Second, in many real-world scenarios, such as extracting long object lists or open-ended facts, knowing what has been missed is as critical as what has been extracted. This brings forth the challenge of *defining completeness*: "What does it mean to extract *all* relevant objects, especially when the ground truth is not explicitly known or is inherently incomplete?" Third, *lack of benchmarks* tailored for high-recall IE creates additional challenges for both extraction and comparative evaluation of baselines.

The following key challenges arise in the context of recall-oriented IE systems operating across diverse sources. These are the target of this dissertation.

Extraction from the Web. For a given subject-predicate pair of interest, the web can contain millions of relevant documents. For instance, consider the task of extracting *all the subsidiaries of Alphabet Inc.* One must sift through millions of pages, disambiguate between entities such as *"Google Energy LLC"* and *"Google Energy PVT LTD,"* avoid re-extracting the same fact from different sources, and ensure coverage of region-specific or long-tail entities buried deep in blogs and forums. More concretely, this is challenging due to following factors:

1. *Entity ambiguity.* Entities appear in various surface forms, making entity disambiguation a persistent and open problem due to polysemy and synonymy.

2. *Sparsity and long-tailed facts.* Information about a valid subject-predicate-object triple can be

spread across multiple web documents. Wikipedia is known to contain common and relevant information particularly for popular entities. In contrast, long-tail facts are often present in news articles, discussion forums, blog posts, social media and others, occurring only scarcely.

3. *Duplicate content.* The same fact often appears across multiple web sources. Without prioritizing relevant documents, exhaustively processing syndicated content can be time-consuming, wasting computational resources without boosting recall.

4. *Temporal drift.* Corporate structures, presidencies, and geopolitical boundaries evolve. Static KBs cannot track short-lived facts (e.g., interim appointments), and existing IE methods struggle with temporal information, such as facts that are true only within specific time boundaries.

Extraction with LLMs. Large language models excel at retrieving single facts but falter when probed for multi-valued relations (e.g., retrieving all instruments played by *Mike Oldfield* through the prompt, "Mike Oldfield played the `[MASK]`."). The token probabilities are skewed by training data frequencies, and cloze-style prompts only reveal top-ranked items. Specifically, the following challenges arise when directly extracting from LLMs' parametric knowledge:

1. *Unknown cardinality.* When the number of correct objects for a given multi-valued subject–predicate pair is unknown, extracting all of them becomes inherently difficult. Cardinality can vary drastically across predicates—e.g., two for `parents` versus more than 100 for `subsidiaries` of a company. Even within a single predicate, it can widely vary by subject—e.g., 100+ subsidiaries for a major corporation versus fewer than 10 for an early-stage startup.

2. *Uncalibrated likelihood.* When probing an LLM with explicit mentions of a subject-predicate pair, the token distribution remains uncalibrated, favoring popular (common) entities. Rarer but valid items may rank far down the distribution, making the extraction process non-trivial.

3. *Popularity bias.* Using autoregressive models to freely perform next-token generation for a given subject-predicate pair, yields the most salient objects first, then drifts to other, often doubtful entities.

4. *Prompt sensitivity.* The structure of the prompt has a direct effect on the token distribution. This leads to inconsistent extractions, undermining reliability.

Extraction from Long Documents. With scaling laws and improvements in language modeling, common factual knowledge can now be easily generated using LLMs. However, domain-specific sources—such as books, legal codes, and scientific reports—which embed hundreds of valid facts across thousands of pages, remain challenging. The length of such documents often exceeds a single LLMs' context window, and processing them with long-context models incurs higher runtime costs and reduced traceability. Moreover, characters and terms frequently reappear under aliases or pronouns, and subtle narrative cues may define relationships. Key obstacles in IE from long documents are:

1. *Long context modeling.* Consuming long documents in a single prompt is not possible under current model constraints. This can be addressed by using in-context learning with document chunks augmented in the prompt for RAG. However, for IE, it is hard to extract from chunks when dependencies between entities can be long-range (across chapters or sections).

2. *Entity resolution.* Entities appear under different names, nicknames, and pronouns, making entity resolution non-trivial. Current knowledge repositories do not contain exhaustive alias lists.

3. *Subtle semantics.* Narrative texts often convey predicates through implicit or context-dependent cues, making them difficult to detect and extract reliably.

4. *Hallucination control.* LLMs blend parametric knowledge with context-specific mentions, leading to hallucinated outputs. Detecting and pruning false positives, especially for long-tail entities, remains a challenge.

1.4 Thesis Contributions

This dissertation develops novel frameworks for high-recall information extraction (IE). Using advanced language modeling capabilities, we integrate principles from IE, information retrieval, and NLP to improve the coverage of knowledge extraction. To make high-recall extraction more tractable, we reframe the problem into solvable sub-tasks, tackle the above mentioned challenges, and aim for completeness by systematically performing exhaustive extraction, robustly handling ambiguity and boundary cases, and identifying long-tail entities and multi-valued relations.

Quantifying coverage and extraction from the web. We first define the notion of coverage and propose the task of predicting information coverage for web documents (Chapter 3). Coverage measures, for a given subject-predicate pair, the fraction of entities extracted by a relation extraction method relative to the ground truth. We introduce HERB, a lightweight classifier that predicts coverage using document- and source-level features, enabling efficient document prioritization in IE pipelines. Using HERB, state-of-the-art relation extraction systems achieve a substantial increase in extraction yield under fixed computational budgets.

Handling multi-valued relations and extraction from LLMs. We uncover calibration issues in LLMs when probing for multiple facts in a single prompt (Chapter 4). To address this, we propose diverse prompt ensembles to elicit multiple answers from LLMs' parametric knowledge. We first generate ranked candidate objects via cloze-style prompts, and then apply relation-specific sampling strategies (e.g., top-k and cumulative probability thresholds) to select multiple entities. Unlike prior methods, we maintain a broad pool of candidate objects, use an ensemble of domain-specific prompts, and train an F1-optimized selector model to filter false positives while striving for recall.

Iterative prompting and extraction from long documents. We introduce the first book-length IE system, L3X, a two-stage framework for long object list extraction (Chapter 5). Stage 1 performs recall-oriented generation via retrieval-augmented generation with iterative prompting to overcome context window limitations. Stage 2 performs precision-oriented scrutinization by employing a self-supervised classifier to prune false positives. L3X substantially improves recall over strong baselines, with particular gains for long-tail entities, difficult relations, and books underrepresented in LLM pretraining.

Capturing temporal knowledge and performing faithful generation. To capture evolving facts and fresh content on the web, we develop Neon, which extracts OpenIE-style propositions from news streams (Chapter 6). These propositions enable LLMs to generate factually relevant and temporally grounded answers to telegraphic user queries. We also explore using LLM-as-a-Judge to evaluate generated responses, revealing both the potential for scalable evaluation and the challenges of maintaining consistency and statistical rigor.

1.5 Publications

The results and contributions of this dissertation have been presented by the first author in the following publications.

1. (Singhania et al., 2022b) Sneha Singhania, Simon Razniewski, and Gerhard Weikum. "Predicting Document Coverage for Relation Extraction." In: *Transactions of the Association for Computational Linguistics*, TACL, 2022. (Chapter 3)

2. (Singhania et al., 2023a) Sneha Singhania, Simon Razniewski, and Gerhard Weikum. "Extracting Multi-valued Relations from Language Models." In: *Proceedings of the 8th Workshop on Representation Learning for NLP*, RepL4NLP at ACL, 2023. (Chapter 4)

3. (Singhania et al., 2025b) Sneha Singhania, Simon Razniewski, and Gerhard Weikum. "Recall Them All: Retrieval-Augmented Language Models for Long Object List Extraction from Long Documents." In: *Proceedings of the 1st Workshop on Natural Language Processing and Language Models for Digital Humanities*, CLARIN Workshop at RANLP, 2025. (Chapter 5)

4. (Singhania et al., 2025c) Sneha Singhania, Simon Razniewski, and Gerhard Weikum. "L3X: Long Object List Extraction from Long Documents." In: *Demo Track Proceedings of the ACM International Conference on Information and Knowledge Management*, CIKM, 2025. (Chapter 5)

5. (Singhania et al., 2025a) Sneha Singhania, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. "Neon: News Entity-Interaction Extraction for Enhanced Question Answering." In: *Proceedings of the 1st Workshop on Robust Information Retrieval*, RobustIR at SIGIR, 2025. (Chapter 6)

Other Publications. The author of this thesis has also co-authored the following papers, reports, and a survey, related to the theme of this dissertation but not included in this text.

1. (Singhania et al., 2022a) Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski. "LM-KBC: Knowledge base construction from pre-trained language models.". In: *The Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models*, CEUR Workshop Proceedings, 2022.

2. (Razniewski et al., 2023) Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, and Jeff Z. Pan. "LM-KBC: Knowledge base construction from pre-trained language models.". In: *The Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models*, CEUR Workshop Proceedings, 2023.

3. (Veseli et al., 2023b) Blerta Veseli, Sneha Singhania, Simon Razniewski, and Gerhard Weikum. "Evaluating Language Models for Knowledge Base Completion." In: *The Semantic Web 20th International Conference*, ESWC, 2023.

4. (Pan et al., 2023) Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Ste- fan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. "Large Language Models and Knowledge Graphs: Opportunities and Challenges". In: *Transactions on Graph Data and Knowledge*, TGDK, 2023.

5. (Razniewski et al., 2024) Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jeff Z. Pan, Tuan-Phong Nguyen, and Bohui Zhang. "LM-KBC: Knowledge base construction from pre-trained language models.". In: *The Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models*, CEUR Workshop Proceedings, 2024.

1.6 Outline

Chapter 2 provides background on information extraction (IE) and reviews existing work on large-scale extraction methods and their downstream applications. The main contributions on IE from the web are presented in Chapters 3 and 6, IE with LLMs in Chapter 4, and IE from long documents in Chapter 5. Finally, Chapter 7 offers concluding remarks, summarizing lessons learned from each project, outlining the limitations of the proposed methods, and highlighting potential future directions.

2

Background

This chapter provides background on information extraction (IE) in Section 2.1, an overview of prior IE pipelines operating at web scale in Section 2.2, IE with large language models in Section 2.3, and IE for long-form documents in Section 2.4. Finally, important downstream applications and tools for each of these areas are covered in Section 2.5.

2.1 Information Extraction

Information extraction (IE) is a fundamental task within natural language processing (NLP) that focuses on extracting structured information from unstructured and semi-structured sources (Han et al., 2020; Xu et al., 2024a; Zhao et al., 2024b). These sources include human-generated texts, such as reports, web documents, social media posts, literary works, and news articles. Content generated by artificial intelligence systems can also serve as input. The goal of IE is to transform free text into a structured format that is readily machine-readable and computable.

The output of IE system is a set of structured, relational tuples connecting entities (Lu et al., 2022b). A common representation is the triple format (`subject-entity`, `predicate`, `object-entity`), encoding a factual statement. For example, from the text, "*MP3 audio format was invented at the Fraunhofer Institute*", an IE system could extract the triple (`MP3`, `developedBy`, `Fraunhofer Institute`), linking the subject *MP3* and the object *Fraunhofer Institute* by the predicate `developedBy`. Such structured facts are also referred to as relational facts.

IE can extract facts that are stated explicitly (as in the MP3 example) as well as facts that are implicit and require inference or world knowledge. For instance, given the sentence "*Saarland University is a public research university located in Saarbrücken, Germany.*", an IE system can infer the fact (`Germany`, `contains`, `Saarbrücken`), since Saarbrücken is a city in Germany, even though the sentence does not state the country-city containment relation explicitly. In essence, IE imposes semantic structure on free text. This transformation is useful as a preliminary stage in larger information processing pipelines. It enables machines to reason and populate knowledge bases for downstream usage—all those operations that would be far more difficult with raw text alone.

2.1.1 Core Components

Information extraction is not a monolithic task but a pipeline of several sub-tasks, discussed below.

2.1.1.1 Named-Entity Recognition

Named-Entity Recognition (NER) identifies and categorizes key information, or "named entities," within text. These entities are predefined categories such as persons, organizations, locations, dates, monetary values, and others.

Definition: Given an input sequence of tokens $S = (s_1, s_2, \ldots, s_n)$, NER identifies spans of text $(s_i \ldots s_j)$ that constitute a named entity and assigns each span a label or category, e.g., PERSON, , ORG, or GPE (for geopolitical entities: countries, cities, or states).

Example: Consider the sentence: "MP3 audio format was invented at the Fraunhofer Institute under the lead of Karlheinz Brandenburg." An NER tool from SpaCy[1] (Honnibal et al., 2020) would identify the following entities and their corresponding tags:

- "MP3" as `PRODUCT`
- "Fraunhofer Institute" as `ORG`
- "Karlheinz Brandenburg" as `PERSON`

The Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996) introduced NER as a standalone task, with only three categories: persons, organizations, and locations. The task gained further popularity under CoNLL-2002 (Tjong Kim Sang, 2002) and CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) tasks. Early NER systems relied on hand-crafted rules, gazetteers, and regular expressions combined with classical sequence models driven by sparse, task-specific features, and were developed for multiple languages (Nadeau and Sekine, 2007). However, these approaches were brittle under domain shift and struggled with challenges such as out-of-vocabulary tokens, nested or discontinuous entities, and long-range context.

With advances in neural models, NER systems relied on feature engineering by leveraging distributed representations (Chiu and Nichols, 2016; Panchendrarajan and Amaresan, 2018; Yadav and Bethard, 2018). As a sequence tagging task within NLP, NER gained further traction (Huang et al., 2015; Reimers and Gurevych, 2017), covering a broader set of categories. The progress was amplified by contextual encoders such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019), which enabled span-based formulations that extended naturally to nested NER. Today, instruction-tuned language models can perform few-shot and zero-shot NER (Xie et al., 2023; Wang et al., 2025), and even adapt to novel schema through prompt-based methods (Ma et al., 2022). Recent surveys on NER are given by Jehangir et al. (2023); Keraghel et al. (2024).

Beyond IE, NER continues to serve as a valuable preprocessing step in a range of NLP applications, most notably in question answering and machine translation.

2.1.1.2 Entity Resolution

Entity-centric understanding in IE hinges on three closely related tasks: *entity linking* (EL), *named-entity disambiguation* (NED), and *coreference resolution* (CR). EL connects textual mentions to canonical entries in a knowledge base (KB). NED is often used interchangeably with EL, or more narrowly to denote the disambiguation step after candidate generation, where the goal is to resolve a mention's ambiguity to its correct KB entry. CR, in contrast, clusters mentions that refer to the same real-world entity, or even concept, without requiring a KB. These components prevent fragmented representations of entities across and within documents.

[1] https://spacy.io/usage/linguistic-features#named-entities

Definitions:

- EL/NED: Given an entity mention in a text, the task is to link it to the correct KB entry or assign Nil if absent. NED specifically handles the step of resolving ambiguity among candidates.
- CR: Given a text, the task is to cluster mentions (e.g., names, pronouns, noun phrases) that refer to the same entity, forming a coreference chain.

Example: For the text "Apple released its new iPhone. It features a faster processor.", EL/NED and CR would produce the following outputs:

- EL/NED connects "Apple" → "Apple Inc." (company entity) and "iPhone" → "iPhone" (product KB entry). If a new unreleased model lacks a KB page, EL assigns Nil and clusters its mentions.
- CR links "It" → "iPhone" and clusters all mentions of the phone.

EL/NED is decomposed into three stages: (i) *entity recognition* to identify and classify mentions; (ii) *candidate generation* to retrieve plausible KB entities using entity lexicons and retrieval; and (iii) *disambiguation* to select the correct entity from candidates (Bunescu and Paşca, 2006; Cucerzan, 2007; Rao et al., 2013; Shen et al., 2015). Early systems combined hand-crafted features, string matching, and local/global statistical models (Ratinov et al., 2011; Medelyan et al., 2009; Dredze et al., 2010). With the rise of knowledge graphs and linked data, graph-based methods using entity-entity relations improved disambiguation accuracy (Hoffart et al., 2011; Al-Moslmi et al., 2020). Neural methods learned representations for mentions and entities in shared spaces, enabling efficient retrieval and context-aware re-ranking (Moro et al., 2014; Ling et al., 2015; Yamada et al., 2016; Sevgili et al., 2022). Community resources such as `GERBIL` standardized evaluation across datasets and measures (Usbeck et al., 2015), while `TAC-KBP` tracks (Ellis et al., 2016; Ji et al., 2017) formalized end-to-end linking (mention detection + linking + Nil-clustering).

Subsequently, dense-retrieval linkers retrieve candidates with a bi-encoder and re-rank them with a cross-encoder, achieving strong accuracy-latency trade-offs and zero-shot generalization (Gillick et al., 2019; Wu et al., 2020). End-to-end training that jointly learns mention detection and disambiguation further reduces error propagation (Kolitsas et al., 2018). Work by Orr et al. (2021) targeted long-tail entities via self-supervision and relational signals. More recently, generative models reframed EL as autoregressive text generation: given context, LLM generates the canonical entity name under constrained decoding (Cao et al., 2021; De Cao et al., 2022). While LLMs' parametric knowledge is helpful, they introduce challenges for EL/NED, including inconsistent outputs and hallucinated or duplicate entity labels, especially for rare entities. Mitigating these errors requires retrieval-augmented generation (RAG) and stricter decoding (Ji et al., 2023; Kandpal et al., 2023).

CR entered the evaluation landscape with the Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996). Early approaches combined deterministic rule-based sieves, applied from high to low precision (Raghunathan et al., 2010; Lee et al., 2011), with feature-rich statistical models. The CoNLL-2012 shared task over OntoNotes was a large, standardized benchmark, creating progress (Pradhan et al., 2012). Machine learning systems then cast CR as mention-pair (Soon et al., 2001) or entity/mention ranking (Denis and Baldridge, 2007) tasks that classified whether two mentions co-refer. However, these models were highly dependent on high-quality linguistic annotations (e.g., parse trees) and were fragile to parsing errors. Later work mitigated these weaknesses via global inference and entity-centric clustering (Clark and Manning, 2015, 2016).

End-to-end neural models replaced manual features with learned span representations, jointly detecting mentions and predicting antecedents at document scale (Lee et al., 2017, 2018). Span-aware pretraining, such as SpanBERT (Joshi et al., 2020), further improved robustness and reduced reliance on external pipelines. Beyond single documents, cross-document coreference and domain generalization have been addressed with models operating over predicted mentions (Cattan et al., 2021; Toshniwal et al., 2021). In parallel, instruction-tuned LLMs have enabled zero- and few-shot coreference via prompting (Le and Ritter, 2024). LLMs' commonsense knowledge has yielded near-human performance on reasoning-focused probes such as the Winograd Schema Challenge (Levesque et al., 2012; Ng, 2017). Nonetheless, LLM-based CR remains highly sensitive to prompt design and inconsistent on long documents (Yang et al., 2022; Zhao et al., 2024a).

2.1.1.3 Relation Extraction

Relation Extraction (RE) identifies and classifies semantic relationships between two or more named entities within text. The set of possible relations is predefined such as `hasMember`, `partOf`, `father`, `mother`, and so on.

Definition: Given a text T and two entities e_1 and e_2, RE determines whether a relation $r \in \mathcal{R}$ holds between them, where $\mathcal{R}$ is a predefined set of relation types. The output is a triplet (e_1, r, e_2).

Example: Consider the sentence: "MP3 audio format was invented at the Fraunhofer Institute under the lead of Karlheinz Brandenburg." Let e_1 be "MP3" and e_2 be either "Fraunhofer Institute" or "Karlheinz Brandenburg." An RE system would extract the following triples:

- (`MP3`, `developedBy`, `Fraunhofer Institute`)
- (`MP3`, `developedBy`, `Karlheinz Brandenburg`)

The RE task was promoted by the Message Understanding Conferences (MUCs) (Sundheim, 1993) and the NIST Automatic Content Extraction (ACE) program (Doddington et al., 2004). Classical RE methods relied on syntactic parse trees (Huffman, 1995; Califf and Mooney, 1997) and feature-based methods (Kambhatla, 2004; Zhou et al., 2005, 2007; Fundel et al., 2006). The highly influential distant supervision technique (Suchanek et al., 2009; Mintz et al., 2009) provided labeled data for training these models. The core idea was to align text with triples from an existing knowledge base (KB), such as Freebase (Bollacker et al., 2008) or Wikidata (Vrandecic, 2012). Specifically, if two entities were known to have a relation in the KB, then all sentences in the underlying corpus that mention both entities were assumed to be positive examples for that relation. While this approach helped scale the dataset, it also introduced noise. In practice, this capped precision and made it difficult for purely pattern- or feature-driven pipelines to generalize.

With the success of distributed word embeddings (Mikolov et al., 2013a,b), neural RE were trained directly from context (Zeng et al., 2014; Zhang et al., 2015; Miwa and Bansal, 2016). Large benchmarks, such as SemEval-2010 Task 8 (Hendrickx et al., 2010), TACRED (Zhang et al., 2017), and DocRED (Yao et al., 2019; Tan et al., 2022), made systematic comparison possible. Transformer-based encoders, especially BERT (Devlin et al., 2019), advanced RE by addressing task-specific challengess: by conditioning each entity mention on both left and right context, they improved disambiguation of roles and relation semantics; by modeling long-distance interactions, they reduced the brittleness of earlier local classifiers in sentences with distant or intervening mentions (Pouran Ben Veyseh et al., 2020). Baldini Soares et al. (2019) showed that simple entity markers over BERT let the model learn task-agnostic relation representations that transfer to unseen entity pairs and infrequent relation types, while a span-level joint model on BERT embeddings (Eberts and Ulges, 2020) mitigated pipeline error by coupling entity and relation decisions. Beyond classification, automatic prompt design for cloze-style formulations and context enrichment, via label-verbalization, retrieval, or knowledge cues (Ye et al., 2022; Chen et al., 2022b,c,a), leveraged the semantics of the label space and improved performance. These gains extend to document-level extraction when augmented with document context and retrieval (Ma et al., 2023b).

BERT-based classifiers assume a closed label set, require per-candidate classification, and struggle with long or cross-sentence contexts. This motivated a shift to seq2seq-style (Lewis et al., 2020a) or autoregressive (Brown et al., 2020) generation, which produces relation tuples directly rather than selecting from a predefined label set (Wadhwa et al., 2023; Li et al., 2023). Crucially, with zero- and few-shot prompting, models can adapt to new schemas and domains without parameter updates (Meng et al., 2022; Han et al., 2022; Wan et al., 2023; Yuan et al., 2023), achieving competitive performance even in specialized clinical RE (Agrawal et al., 2022). This also enable schema-free, evidence-grounded outputs (Zeng et al., 2020; Paolini et al., 2021; Huguet Cabot and Navigli, 2021; Zhou et al., 2021). Recent surveys (Zhao et al., 2024b; Qin et al., 2024) highlight deep learning techniques for RE and outline future directions to address challenges of real-world RE systems.

2.1.1.4 Event Extraction

Event Extraction (EE) detects mentions of events in text and extracts their participants as structured arguments. Since each event type defines a schema of roles (slots) to be filled by entities, EE intersects with the formalism of slot-filling and the general paradigm of semantic role labeling.

Definition: An event is defined by a trigger (often a verb or noun signifying the event) and a set of argument slots (entities). The core task involves detecting the event trigger and classifying all associated arguments. EE is analogous to a structured slot-filling task, where each event schema denotes predefined slots to be filled. This process moves beyond static, binary relationships often found in relation extraction (see Section 2.1.1.3), and captures dynamic situations with multiple participants, essentially answering the "5W1H" questions—who, what, when, where, why, and how—about an occurrence.

Example: In the sentence: "Google announced a deal on Tuesday to buy rapidly growing cybersecurity firm Wiz for $32 billion,"[2] an event extraction system would identify:

[2] https://edition.cnn.com/2025/03/18/tech/google-wiz-acquisition

- Event Trigger: "acquired"
- Event Type: Acquisition
- Arguments:
  - Acquirer: Google (`ORG`)
  - Acquired: Wiz (`ORG`)
  - Price: $32 billion (`MONEY`)
  - Time: Tuesday (`DATE`)

EE was central to the Message Understanding Conference (MUC-6) evaluation (Grishman and Sundheim, 1996) and continued under Automatic Content Extraction (ACE) (Doddington et al., 2004). Early approaches framed extraction as a pipeline: first detect event triggers using lexical and syntactic features, then classify arguments and roles (Ahn, 2006; Ji and Grishman, 2008; Liao and Grishman, 2010). These pipelines scaled, but suffered from error propagation, were brittle for rare event types, and struggled at cross-sentence reasoning. Neural joint models addressed these failure modes by jointly modeling the representations for triggers and arguments (Nguyen and Grishman, 2015; Chen et al., 2015; Nguyen et al., 2016a).

Subsequent work reframed EE as machine reading comprehension (MRC) or question-answering (QA) tasks (Liu et al., 2020; Du and Cardie, 2020; Liu et al., 2021). Lu et al. (2021) treated EE as a generation task with variable argument cardinality, rather than committing to per-candidate classification. Later transformer encoders aggregated long-range context for multi-argument events (Wadden et al., 2019). At document level, building event graphs across sentences by linking coreferent mentions and jointly filling roles proved effective (Zheng et al., 2019a). Finally, large seq2seq and instruction-tuned models formulate EE as template filling task, adapting to new ontologies (Lu et al., 2022b) and low-supervision settings (Hsu et al., 2022). With generative models, constrained decoding injects schema constraints at inference time (Lu et al., 2021), and retrieval-augmented prompting supports document-level argument filling (Li et al., 2022b; Ren et al., 2023).

Parallel to but somewhat distinct from EE, the slot-filling task for IE and KB population frames the problem as filling information slots (e.g. "founder of X"). In such systems, one issues a query containing a head entity and slot/blank, and then extracts the appropriate filler (tail) from candidate text spans (Louvan and Magnini, 2020; Weld et al., 2022). This approach is used in the TAC-KBP slot filling challenge (Surdeanu, 2013). Beyond slot-filling, semantic role labeling (SRL) offers a deeper connection to EE because both tasks involve predicate-argument structure. SRL seeks to label general semantic roles for predicates across sentences (Palmer et al., 2005; Màrquez et al., 2008; Palmer et al., 2013; He et al., 2015, 2017, 2018; Shi and Lin, 2019). Work by Zhang et al. (2022) explicitly studies transfer from SRL to event argument extraction.

2.1.2 Metrics and Evaluation

To evaluate the performance of Information Extraction (IE) systems, we use standard classification metrics that measure how well the system's predictions match a set of "gold" or ground-truth annotations. The fundamental metrics are Precision, Recall, and the F1 score, which are calculated based on the counts of True Positives (TP), False Positives (FP), and False Negatives (FN).

- True Positive (TP): A correct prediction made by the system.
- False Positive (FP): An incorrect prediction made by the system.
- False Negative (FN): A correct item that the system failed to predict.

Precision measures the accuracy of the predictions. It answers the question: "Of all the items the system predicted, how many were actually correct?" High precision means the system makes few false positive (FP) errors.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.1}$$

Recall measures the completeness of the predictions. It answers the question: "Of all the items that should have been predicted, how many did the system find?" High recall means the system makes few false negative (FN) errors.

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.2}$$

F_1 score is the harmonic mean of Precision and Recall, providing a single score that balances both metrics. It is useful in finding a compromise between making few mistakes (precision) and finding all correct instances (recall).

$$F_1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{2.3}$$

When a task involves multiple classes or types (e.g., different entity types like "Person" and "Organization"), the overall performance can be calculated using different averaging strategies.

micro-F_1: Micro-averaging calculates the metrics globally by summing the instance-level TP, FP, and FN counts across all classes before computing the final score. This emphasizes performance on frequent classes.

$$\text{micro } F_1 = \frac{2 \times \sum \mathrm{TP}_c}{\sum(2 \times \mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c)} \tag{2.4}$$

where the sum is over all classes c.

macro-F_1: Macro-averaging calculates the metric independently for each class and then takes the unweighted average of the scores. This gives equal weight to each class, regardless of how frequent it is. It's a good measure of how well the system performs on rare classes.

$$\text{macro } F_1 = \frac{1}{|C|} \sum_{c \in C} F_{1c} \tag{2.5}$$

where C is the set of all classes and F_{1c} is the F_1 score for the class c.

In the context of information extraction, these metrics are applied to evaluate the structured outputs produced by its core sub-tasks. A *prediction* is considered correct only if it exactly matches the gold annotation. In each of the core components, the evaluation would be as follows:

- Named Entity Recognition: A prediction can be (span of text, entity-type) pair, like ("New York", `location`). A true positive requires the system to identify the exact or overlapping ("New York City" for "New York") text span and its correct type.

- Entity Resolution: For entity linking/named-entity disambiguation each identified entity mentions must link to the correct KB entry (for e.g., "NYC" gets linked to "New York City" entry in Wikidata) and performance is measured as mention-level accuracy (with optional precision/recall for NIL detection). For coreference resolution, mentions referring to the same KB entity must be clustered together and evaluation commonly reports F_1.

- Relation Extraction: A prediction is a triplet, such as (entity$_1$, relation-type, entity$_2$), like ("Berlin", capitalOf, "Germany"). A true positive requires that the two entities and the relation connecting them are all correct.

- Event Extraction: Scoring is often two-fold. First, for identifying event triggers (the word indicating an event), a true positive is a correctly identified (span, event-type). Second, for argument role labeling, a true positive is a correctly identified (event trigger, argument, role) triple, such as ("elected", "Angela Merkel", personElected).

When the output is in the form of ranked list, which is often the case for multi-valued relations or search results from ranking systems, the following metrics can be used for evaluation.

nDCG: When outputs are ranked and relevance can be graded (e.g., {0,1} for match with ground-truth), we use normalized Discounted Cumulative Gain (nDCG) (Järvelin and Kekäläinen, 2002). Let r_i be the relevance label of the item at rank i and $g(r)$ a gain function (commonly $g(r) = 2^r - 1$). The discounted cumulative gain at cutoff k is

$$\mathrm{DCG@}k = \sum_{i=1}^{k} \frac{g(r_i)}{\log_2(i+1)}.$$

Let IDCG@k be the DCG of the ideal ranking (items sorted by r_i). Then

$$\mathrm{nDCG@}k = \frac{\mathrm{DCG@}k}{\mathrm{IDCG@}k} \in [0,1],$$

which normalizes for list difficulty and enables comparison across queries. nDCG enables fair comparisons between different lists, regardless of their length and the scale of relevance scores.

P@R: Precision at a target recall level measures the best precision achievable while covering at least a fraction R_0 of the gold items. Given a scoring function and threshold τ,

$$\text{P@R}_{R_0} \;=\; \max_{\tau}\ \text{Precision}(\tau) \quad \text{such that} \quad \text{Recall}(\tau) \;\geq\; R_0.$$

For ranked lists, the equivalent is to choose the smallest cutoff k whose cumulative recall meets R_0 and report the corresponding precision.

R@P: Recall at a target precision reports the maximum recall attainable while maintaining at least precision P_0:

$$\text{R@P}_{P_0} \;=\; \max_{\tau}\ \text{Recall}(\tau) \quad \text{such that} \quad \text{Precision}(\tau) \;\geq\; P_0.$$

Both P@R and R@P summarize operating points on the precision-recall curve that are often of practical interest (e.g., "retrieve 90% of facts with at least 70% precision").

AUPRC: Area under precision-recall curve focuses on how well the model ranks relevant items above irrelevant ones within the generated lists. This is an adapted version of area under curve, used for classification tasks. In the discrete form, this coincides with Average Precision (AP):

$$\text{AP} \;=\; \sum_{k=1}^{N} \text{P@}k \cdot \Delta\text{R@}k \;=\; \frac{1}{|G|} \sum_{k=1}^{N} \text{P@}k\, \mathbf{1}\{y_k = 1\},$$

where $y_k \in \{0,1\}$ indicates whether the item at rank k is relevant, $|G|$ is the number of ground-truth items, and $\Delta\text{R@}k$ is the recall increment at rank k.

2.1.3 High-coverage Information Extraction

Information extraction (IE) entails an inherent precision-recall trade-off. In many downstream settings such as knowledge base construction, scientific discovery, and evidence retrieval, the cost of a false negative (a missed true fact) exceeds that of a false positive (a spurious candidate that can be filtered later). This dissertation advances large-scale IE frameworks with an *explicit emphasis on recall*. In our formulation, the IE task is cast in terms of the standard relation extraction setup or its variants, since identifying relations is the most crucial and challenging aspect of IE, requiring models to accurately capture the semantics of the given context. Other core IE components are addressed implicitly through input pre-processing or neural models themselves.

The main contributions of this thesis are organized into three themes, distinguished by the source of input: the web, large language models, and long documents. While prior work was summarized in Sections 1.2 and 2.1.1.3, this chapter provides a more detailed background for each of these domains in Sections 2.2, 2.3, and 2.4.

2.2 Information Extraction from the Web

Web-scale IE focuses on extracting facts from large, heterogeneous text collected from the web. The development of various RE methods under this theme is discussed below.

2.2.1 Pattern Extraction Models

Early approaches of web-scale RE relied on scalable pattern mining (Huffman, 1995; Soderland et al., 1995; Kim and Moldovan, 1995; Califf and Mooney, 1997). These methods performed iterative pattern learning by starting with a few seed examples (i.e., a dictionary of patterns) to induce new textual patterns and extract additional facts. Brin (1998) introduced the *DIPRE* method, which exploits the duality between sets of patterns and relations to grow a target relation from a small sample. Building on this idea, *Snowball* (Agichtein and Gravano, 2000) presented a bootstrapping approach for RE from plain-text documents, using seed patterns and clustering-based confidence measures to automatically identify "sufficiently reliable" tuples and patterns for subsequent iterations (a snowballing effect). Later work extended this to larger and diverse corpora such as news, tweets, and scientific papers, with a focus on improving both accuracy and coverage (Carlson et al., 2010; Jiang et al., 2017), scaling the expressive power of patterns (Etzioni et al., 2004; Nakashole et al., 2012), and enhancing pattern quality (Zheng et al., 2019b). Despite these advances, pattern-based methods suffered from systematic errors and required human validation of extracted patterns.

2.2.2 Statistical and Neural Extraction Models

Conventional statistical (Zelenko et al., 2002; Zhou et al., 2005) and neural (Zeng et al., 2014; dos Santos et al., 2015) methods treat RE as a supervised classification task that predicts the relation type between two entities mentioned in a sentence. Feature-based approaches design lexical, syntactic, and semantic features (Kambhatla, 2004; Zhou et al., 2005; Jiang and Zhai, 2007; Nguyen et al., 2007) for entity pairs and their surrounding context, and use these to train classifiers. The training task maps feature vectors to outputs, which may take the form of relation class labels, regression scores, or latent representations such as cluster IDs or embeddings.

In statistical relational learning, entity representations explicitly encode relationships to other entities, yielding graph-structured data. This is commonly referred to as knowledge graphs or knowledge bases, where nodes are entities and edges are relations. It employs probabilistic graphical models to represent and reason about domains with complex structure, with core goals as predicting missing edges, inferring node properties, and clustering entities (Nickel et al., 2016; Fu et al., 2019).

2.2.2.1 Knowledge Bases

Distant supervision for RE made statistical machine learning integral to web IE (Suchanek et al., 2009; Mintz et al., 2009). In this paradigm, existing KBs—such as YAGO (Suchanek et al., 2007), DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), NELL (Carlson et al., 2010), and the Google Knowledge Graph (Singhal, 2012)—are used to label training instances by aligning known facts with sentences that mention the corresponding entity pairs. This eliminates the need for manual annotation and enables learning from massive, heterogeneous web corpora. However, these automatically generated labels are inherently noisy, prompting follow-up work on noise reduction and multi-instance formulations (Riedel et al., 2010; Hoffmann et al., 2011; Takamatsu et al., 2012; Surdeanu et al., 2012; Min et al., 2013; Lin et al., 2016). Overall, coupling text with a KB via distant supervision (schema-closed RE) broadened coverage and enabled robust web-scale extraction.

2.2.2.2 Open Information Extraction

Schema-closed RE scales supervision but under fixed ontologies. Open Information Extraction (OpenIE) addresses this by extracting tuples directly from free text, as ($\texttt{argument}_1$, $\texttt{relation-phrase}$, `argument`$_2$), without committing to a predefined schema. Early web-scale systems showed that shallow syntax, clause decomposition, and precision filters yield high-recall yet reasonably accurate extractions (Yates et al., 2007; Wu and Weld, 2010; Fader et al., 2011; Mausam et al., 2012; Del Corro and Gemulla, 2013; Angeli et al., 2015). Benchmarks such as OIE2016 (Stanovsky and Dagan, 2016) and CaRB (Bhardwaj et al., 2019) standardized evaluation and exposed recurring failure modes, such as overly specific relation phrases and incomplete arguments. Subsequent supervised and neural formulations improved boundary detection, argument completeness, and calibration (Stanovsky et al., 2018; Cui et al., 2018; Kolluru et al., 2020b).

Although OpenIE offers high-recall and domain independence, its unsupervised nature introduces several limitations (Schneider et al., 2017; Niklaus et al., 2018). It can lead to extraction of low-quality, redundant, or tautological tuples, and complex sentence structure can produce incomplete or fragmented arguments. Moreover, since OpenIE forgoes canonicalization, the same relation can be expressed by several surface forms (e.g., "was born in," "is the birthplace of"). This makes post-hoc deduplication and alignment with canonical ontologies difficult. Finally, most systems operate at the sentence level. Recent progress is surveyed by Zhou et al. (2022); Pai et al. (2024).

2.2.2.3 Evidence Retrieval and Document Ranking

At web scale, extraction quality hinges on retrieving the right contexts before modeling. Practical pipelines therefore interleave information retrieval (IR) with IE: they index crawled pages; generate candidates using entity-aware queries, alias expansions, and co-mention or pattern cues; and then combine lexical retrieval with neural re-ranking to prioritize sentences most likely to express a target relation (e.g., BERT-based cross-encoders and re-rankers (Nogueira and Cho, 2019; Khattab and Zaharia, 2020)). In distantly supervised settings, retrieved sentences form a "bag" per entity pair, and multi-instance learning aggregates evidence under label noise (Riedel et al., 2010; Surdeanu et al., 2012; Lin et al., 2016). Post-extraction, systems combine signals across sources and calibrate confidence, sometimes combining scores with KB priors (e.g., *Knowledge Vault*) to balance the final precision/recall scores (Dong et al., 2014). Recent work couples retrieval directly with document-level RE (Ma et al., 2023b). This retrieval–extraction–aggregation loop mitigates noise and drift while focusing on high-yield evidence.

2.2.3 Further Advances and Discussion

Pattern-driven bootstrapping established scalable web IE, but modern pipelines rely on neural models that cast RE as supervised classification, integrate text with KB structure via distant supervision, and employ IR to retrieve high-yield evidence. Since the work in Chapter 3, IE has been reframed as a unified framework for text-to-structure generation (Lu et al., 2022b; Fei et al., 2022). Instruct-tuned models have further improved zero-/few-shot extraction (Wang et al., 2023b; Jiao et al., 2023). In parallel, evidence- and retrieval-augmented document-level RE has strengthened grounding and long-context reasoning (Ma et al., 2023b; Gao et al., 2024). These advances connect to Chapter 4, which studies probing language models directly for multi-valued RE.

2.3 Information Extraction with Language Models

IE from or with language models (LMs) treats pre-trained LMs as knowledge repositories and queries them directly to elicit factual information. The development of this paradigm, and the various methods explored under it, are discussed below.

2.3.1 Language Models

Language models (LMs) are trained to estimate the probability distribution of text sequences using self-supervised learning (Liu et al., 2023a; Xu et al., 2024a). Given a sequence of tokens $t = (t_1, t_2, \ldots, t_n)$, an autoregressive LM like GPT-3 (Brown et al., 2020) factorizes this distribution as

$$P_{\text{LM}}(t; \theta) = \prod_{i=1}^{n} P(t_i \mid t_1, \ldots, t_{i-1}). \tag{2.6}$$

Different architectures vary in how they are trained: autoregressive (e.g., GPT-2 (Radford et al., 2019)), bidirectional (e.g., BERT (Devlin et al., 2019)), or encoder-decoder (e.g., BART (Lewis et al., 2020a)), but all are trained on massive text corpora (Radford and Narasimhan, 2018). In masked LMs, special tokens such as "`[MASK]`" are introduced, and the model is trained to predict the masked token $t_i \in \mathcal{V}$ using its surrounding context:

$$P_{\text{LM}}(t_i \mid t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n; \theta), \quad t_i \in \mathcal{V}. \tag{2.7}$$

During pre-training, the LM's parameters θ are optimized to maximize the likelihood of the underlying corpora. Because these corpora contain numerous factual assertions, LMs implicitly acquire structured, KB-like information alongside syntactic, semantic, and commonsense knowledge (Xiong et al., 2020; Yasunaga et al., 2022; Minaee et al., 2024).

2.3.2 Prompting

Prompting is the technique of querying an LM with a textual template such that the model completes or fills in the prompt to produce the desired output. Formally, for an input x (e.g., a subject entity or a question), a template function f_{prompt} produces a prompt $x' = f_{\text{prompt}}(x)$ that contains one or more blanks or slots to be filled (Liu et al., 2023a). For example, to query a fact such as (`Beethoven`, `bornIn`, ?), one can use the template "[X] was born in [Z]." Filling [X] with "Ludwig van Beethoven," the LM predicts the answer slot [Z] ("Bonn") by exploiting its learned distribution over text (Petroni et al., 2019). In general, the model selects the most probable completion:

$$\hat{z} = \arg\max_{z \in Z} P_{\text{LM}}\big(f_{\text{fill}}(x', z); \theta\big) \tag{2.8}$$

where $f_{\text{fill}}(x', z)$ inserts candidate z into the prompt and Z is the set of admissible answer strings (e.g., token sequences over the vocabulary $\mathcal{V}$). In practice, decoding can be deterministic (e.g., greedy or beam search) for reproducibility, or stochastic (e.g., nucleus/top-p sampling) when diversity across plausible completions is useful (Liu et al., 2023a; Holtzman et al., 2020). For classification tasks, the surface form $\hat{z}$ is mapped to a task label $\hat{y}$ via a *verbalizer*, which defines a label-to-word mapping—a design choice shown to be important in few-shot learning (Schick and Schütze, 2021a; Gao et al., 2021).

Prompting thus reformulates queries or tasks into a format that an LM can naturally handle. Two commonly used prompt types are:

- **Cloze prompt**: A template with a blank in the middle of a sentence, as in the Beethoven example above, where the model predicts a masked token.

- **Prefix prompt**: A template where the input text is given first and the model is prompted to continue, e.g.,"*Question: Ludwig van Beethoven was born in which city? Answer:*" or "*Ludwig van Beethoven was born in*". Here the model generated the answer after the prompt.

Designing effective prompts, termed as prompt engineering, is critical. Early work relied on manual templates based on linguistic patterns (Jiang et al., 2020b). Subsequent research explored automatic prompt generation, either by discrete search or by gradient-based optimization (Shin et al., 2020; Zhong et al., 2021; Qin and Eisner, 2021; Lester et al., 2021; Liu et al., 2022b). Learned prompts, or *prompt tuning*, improved factual recall on benchmarks such as `LAMA` (Petroni et al., 2019), `KILT` (Petroni et al., 2021) and `X-FACTR` (Jiang et al., 2020a), though they may not be human-readable, sometimes containing unnatural token sequences. Overall, prompting allows LMs to "fill-in-the-blank", turning the language modeling into a question-answering task. Recent surveys are given by Liu et al. (2023a); Schulhoff et al. (2024); Sahoo et al. (2024).

2.3.3 Probing

While prompting elicits answers from an LM, probing measures the information encoded within them (Hewitt and Liang, 2019; Talmor et al., 2020). For factual knowledge, probing involves prompting an LM (without additional training) to quantify how much factual knowledge is captured in its parametric memory (a.k.a knowledge probes) (Petroni et al., 2019; Alivanistos et al., 2022).

Probing studies have also shown that results can be highly sensitive to prompt phrasing (Elazar et al., 2021). Even minor rewordings can lead to large differences in whether the model recalls a fact. Ensuring consistency across phrasings, or designing prompts that are robust to variation, remains an ongoing challenge (Schick and Schütze, 2021a; Tam et al., 2021; Lu et al., 2022a). Beyond cloze-style probes, other approaches involve probe classifiers: small models trained on top of an LM's internal representations to predict specific properties (Schick and Schütze, 2021b). Such diagnostic probes test whether LM embeddings encode syntactic roles, semantic categories, or factual attributes (Ettinger, 2020; Kassner and Schütze, 2020; Sun et al., 2021). Recent surveys are given by Youssef et al. (2023); Zhang et al. (2024b).

2.3.4 In-context Learning

LLMs can be conditioned on a small set of input-output examples in the prompt context, enabling predictions at inference time without parameters updates (Brown et al., 2020). This approach, termed as in-context learning (ICL) or few-shot prompting, suggests that LLMs perform meta-learning during pre-training, enabling adaptation to new tasks. The few-shot demonstrations steer the model toward the desired task format and behavior. ICL performance is highly sensitive to the choice and order of examples, the prompt format, and suffer from label and prior bias (Zhao et al., 2021; Min et al., 2022b; Lu et al., 2022a; Min et al., 2022a). Reasoning-based prompting, such as chain-of-thought which adds the cue "Let's think step by step", improves multi-step problem solving (Wei et al., 2022; Kojima et al., 2022). Retrieving demonstrations that better match the test instance has also been explored (Rubin et al., 2022; Liu et al., 2022a). Recent surveys are provided by Liu et al. (2023a); Dong et al. (2024); Luo et al. (2024).

2.3.5 Language Models as Knowledge Bases

By treating LMs as KBs, Petroni et al. (2019) examined whether LMs encode factual data within its parameters and if they can be queried. Formally, let $\mathcal{E}$ be the set of entities and $\mathcal{R}$ the set of relations; a fact is a triple $(s, r, o) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. To query the LM for the object of $(s, r, \cdot)$, associate each relation r with a cloze-style template f_r that yields a prompt with subject and object slots (e.g., "*[X] and [Z] share a border.*"). Given a subject s, replace [X] with its surface form (e.g., "Germany") and let the LM predict [Z] (e.g., "Poland"). Using Eq. (2.8), we say the LM *knows* the fact if the predicted object $\hat{z}$ (or its mapped label $\hat{y}$) matches the ground truth o (or y). Under this view, the LM performs zero-shot object generation akin to a KB lookup.

2.3.6 Confidence Elicitation

Confidence elicitation quantifies the factuality or uncertainty of LM-generated facts. It can be done by prompting the model to explicitly verbalize its confidence (Kadavath et al., 2022), by analyzing the model's internal probabilities via token log-probabilities (Kuhn et al., 2023; Manakul et al., 2023), or by checking output consistency (Weng et al., 2023; Gero et al., 2023). However, transformers are known to be uncalibrated and benefit from post-hoc methods such as temperature scaling (Desai and Durrett, 2020; Xie et al., 2024; Zhang et al., 2024a). Other techniques leverage external sources (like Wikipedia) to measure and evaluate factuality (Min et al., 2023; Chern et al., 2023). Recent surveys are given by Tian et al. (2023); Wang et al. (2023a); Xiong et al. (2024a); Geng et al. (2024).

2.3.7 Further Advances and Discussion

Language models (LMs) opened a new avenue for IE by implicitly serving as KBs. Various prompting techniques enable factual extraction by framing prompts as fill-in-the-blank queries. However, it faces several limitations: (i) knowledge is not concretely organized or interpretable, (ii) generation is brittle to prompt wording, (iii) single slot prompts do not generate multi-token objects, and (iv) LMs lack the explicit schema and precision of existing KBs (Razniewski et al., 2021; AlKhamissi et al., 2022). The feasibility of materializing KBs directly from LMs remains an open problem. Since the work in Chapter 4, further progress addresses some of these challenges through few-shot learning, editing and tracing parametric knowledge, and reasoning-assisted extraction (Akyurek et al., 2022; Meng et al., 2022; Cohen et al., 2023; Geva et al., 2023; Ma et al., 2023a; Kwak et al., 2024). These advances naturally motivate Chapter 5, which focuses on IE from long documents.

2.4 Information Extraction from Long Documents

IE from long documents targets fact extraction over high-volume text and book-length narratives. The development of RE methods in this setting is discussed below.

2.4.1 Document-level Relation Extraction

Document-level RE extends sentence-level RE by training models to read across multiple sentences, aggregate evidence from repeated mentions, and reason over discourse (Li et al., 2016; Quirk and Poon, 2017). The DocRED benchmark turned setting mainstream by providing large-scale, human-annotated document triples with evidence spans (Yao et al., 2019). Later, Tan et al. (2022) found substantial false negatives in DocRED and released Re-DocRED to strengthen evaluation.

In terms of modeling, graph-based methods build mention/entity graphs and propagate information for multi-hop reasoning (Peng et al., 2017; Nan et al., 2020; Zeng et al., 2020; Xu et al., 2021b). Transformer variants augment pre-trained encoders with task-specific mechanisms such as adaptive thresholds, localized context pooling, or global interaction over entity pairs to handle multi-label, cross-sentence decisions (Zhou et al., 2021; Ma et al., 2018). Moreover, evidence-centric approaches explicitly select rationales before classifying relations, improving grounding and robustness (Tang et al., 2020; Han and Wang, 2020; Zhang et al., 2021; Xie et al., 2022; Xu et al., 2022). Generative LLMs complement these methods by jointly decoding entities and relations (Giorgi et al., 2022).

Despite these advances, applying these RE methods to book-length narratives remains challenging: (i) documents exceed LLM context length; (ii) evidence for a single triple can be sparse and scattered; and (iii) naively encoding an entire book can be computationally expensive. This motivates retrieval-first pipelines: first surface candidate sentences/paragraphs, and then perform retrieval-augmented extraction (Josifoski et al., 2022; Ma et al., 2023b; Gao et al., 2024).

2.4.2 Retrievers

In the long documents setting, retrievers play a key role in selecting relevant passages before extraction or downstream usage. Prior work has focused on retrievers for knowledge-intensive NLP. Sparse retrievers, such as BM25 (Robertson and Zaragoza, 2009), SPLADE (Formal et al., 2021), and UniCOIL (Lin and Ma, 2021), rank documents using lexical overlap and term statistics. Dense retrievers, such as DPR (Karpukhin et al., 2020), ColBERT (Khattab and Zaharia, 2020), ANCE (Xiong et al., 2021), and Contriever (Izacard et al., 2022), capture semantic similarity beyond exact word matches. While the choice of retriever depends on the task, they differ along common axes. Sparse methods are simple and fast, but can miss relevant evidence when queries and texts differ in wording. Dense methods require large training sets (e.g., MS MARCO (Nguyen et al., 2016b)) and can struggle to generalize out-of-domain. In practice, modern systems combine them and formulate a hybrid approach[3,4] to leverage the precision of lexical matching and the recall of semantic matching. For long-document IE, retrieval is crucial because it narrows the input to the most relevant segments, allowing effective extraction under limited LLM context lengths.

[3] https://www.elastic.co/search-labs/blog/hybrid-search-elasticsearch

[4] https://opensearch.org/blog/introducing-reciprocal-rank-fusion-hybrid-search/

2.4.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) integrates the retrieval step with text generation. In a RAG pipeline, a query first retrieves a small set of relevant texts (e.g., Wikipedia articles or web passages); then a generator (often an LLM) reads both the query and the retrieved evidence to produce an answer or extraction (Guu et al., 2020; Lewis et al., 2020b). By coupling LLM's parametric memory with non-parametric memory, RAG aims to achieve higher factual accuracy and coverage. Formally, adapting Eq. (2.8), RAG marginalizes the LM score over retrieved contexts:

$$\hat{z} = \arg\max_{z \in Z} \sum_{c \in \mathcal{N}_k(q,D)} p(c \mid q, D)\, P_{\text{LM}}\big(f_{\text{fill}}(x', c, z); \theta\big), \tag{2.9}$$

where D is the datastore (index), q is the retrieval query, c is a retrieved context (sentence or paragraph), $\mathcal{N}_k(q, D)$ denotes the top-k results, and $p(c \mid q, D)$ is the retriever's normalized score (e.g., a softmax over dense retriever similarities).

RAG systems vary along three axes. (i) *retriever*: sparse vs. dense encoders trained to rank evidence for the query; multi-vector retrievers can increase recall for entity-heavy queries (Karpukhin et al., 2020; Khattab and Zaharia, 2020). (ii) *reader/generator*: sequence vs. token marginalization (RAG-Sequence/RAG-Token), or fusion readers that concatenate and jointly attend to multiple passages (Izacard and Grave, 2021; Lewis et al., 2020b). (iii) *coupling*: tight, model-internal coupling during pre-training/instruction-tuning vs. loose, black-box coupling at inference (Borgeaud et al., 2022; Izacard et al., 2023; Shi et al., 2024b). In-context retrieval interleaves retrieved passages directly into the prompt for black-box LMs (Ram et al., 2023), while active retrieval adaptively issues follow-up queries during generation (Jiang et al., 2023).

RAG grounds answers in retrieved evidence and enhances freshness, since the index can be updated independently. This can be efficient strategy: smaller LMs equipped with retrieval can match or surpass larger models (Borgeaud et al., 2022; Izacard et al., 2023). RAG also enables structured IE through typed queries and constrained decoding (Lewis et al., 2020b; Petroni et al., 2021). However, it also introduces challenges, including recall-latency trade-offs in retrieval, evidence selection under distribution shift, and inference over multiple passages.

2.4.4 Long-context Models

A complementary approach to RAG is to expand the model's context window to process the entire document in a single pass. Since self-attention scales quadratically in sequence length, naïve long-context prompting is expensive. Recent architectures and training strategies mitigate this by (i) using sparse/local attention (Beltagy et al., 2020; Zaheer et al., 2020), (ii) tailoring encoder-decoder models for long inputs (Guo et al., 2022), (iii) employing IO-efficient attention kernels (Dao et al., 2022), and (iv) using position interpolation to extend models beyond their original context windows (Chen et al., 2023). Recent LLMs demonstrate million-token contexts in practice (Reid et al., 2024).

Even with efficient attention, long-context inference is costly. Two pragmatic strategies help: (a) *prompt compression*, which summarizes inputs prior to inference (Jiang et al., 2024); and (b) *model cascades or routing*, which send easy queries to smaller models and reserve large long-context models for hard cases (Chen et al., 2024). Orthogonally, retrieval-style external memory augments long-range reasoning, including memorizing caches that "learn at inference time" (Wu et al., 2022), nearest-neighbor decoding (Khandelwal et al., 2020; He et al., 2021), top-k state retrieval for book-length inputs (Bertsch et al., 2023), and joint self-retrieval with the LM (Rubin and Berant, 2024).

Comparative studies find trade-offs between long-context LMs and RAG. Long-context LMs exhibit position sensitivity: accuracy drops when evidence lies in the middle of long prompts ("lost in the middle"), and longer prompts amplify distractor effects (Li et al., 2024; Shi et al., 2023; Wu et al., 2024; Yang et al., 2025). Even with I/O-aware kernels, latency and memory footprints (e.g., KV caches) grow with input length, making million-token prompts expensive. Many long-context benchmarks also over-rely on synthetic "needle-in-a-haystack" setups, obscuring where models truly fail (Kamradt, 2023; Bai et al., 2024). In contrast, RAG is more cost-efficient, scalable, and easier to ground—albeit at the expense of increased LLM calls (Xu et al., 2024b; Li et al., 2025b).

2.4.5 IE from Narrative Texts

Long documents such as novels, books, and literary works, pose unique challenges for IE. Unlike self-contained news or Wikipedia articles, narratives are lengthy, entities recur under aliases and forms, coreference links can span chapters, relations evolve over time, events may be narrated out of order, and speaker attribution interacts with point of view and discourse structure. Effective IE systems must therefore infer at the document-level, linking evidence scattered over many pages.

Research on narratives spans long-form understanding, summarization, and question answering. NarrativeQA introduced book- and script-level comprehension (Kočiský et al., 2018). QuALITY and ChapterBreak probe long-input reasoning and discourse continuity (Pang et al., 2022; Sun et al., 2022). Thai et al. (2022) study literary evidence retrieval with exact supporting quotations. Book-length summarization benchmarks (e.g., BooookScore, FABLES) highlight persistent challenges in faithfulness and content selection (Chang et al., 2024b; Kim et al., 2024), while NoCha stresses book-wide reasoning beyond sentence-level retrieval (Karpinska et al., 2024).

Resources created for literary entities and coreference support much of this work (Bamman et al., 2019, 2020). Building on these datasets, Stammbach et al. (2022) explore passage-level character role labeling (hero/villain/victim) with LLMs, followed by analyses of quote attribution and speaker identification (Chang et al., 2023). More recently, LLMs have been used as classifiers for cultural-analytic features (Bamman et al., 2024) and for discourse-focused analysis of time, setting, and perspective (Piper and Bagga, 2024). Despite this progress, most approaches still optimize for a single span, passage, or label, e.g., character profiling; quote attribution (Yuan et al., 2024; Michel et al., 2025). End-to-end RE from entire books remains underexplored.

2.4.6 Further Advances and Discussion

IE from long documents, especially narrative text, requires reasoning over non-local evidence: entities recur under aliases, coreference spans chapters, relations evolve temporally, and key clues are scattered across the book. Document-level RE makes this setting explicit by annotating cross-sentence triples and evidence (Yao et al., 2019). To scale beyond encoder limits, retrieval narrows inputs to salient passages, and RAG conditions decoding on these passages to improve faithfulness (Guu et al., 2020; Lewis et al., 2020b). Narrative benchmarks underscore the need for book-level understanding, highlighting the gap between passage-level supervision and end-to-end RE across entire books (Kočiský et al., 2018; Bamman et al., 2019, 2020; Chang et al., 2023).

2.5 Applications

Information Extraction (IE) supports a wide range of downstream tasks including search, knowledge base construction, fact-checking, and question answering (QA). The application of IE system in each of theses is discussed below.

2.5.1 Search engines

IE plays an important role in powering search engines. Traditionally, search was characterized by the "ten blue links" model using lexical (sparse) information retrieval (IR) techniques (Manning et al., 2008). In this model, the search engine matched query terms against an inverted index and ranked documents by relevance signals (e.g., using TF-IDF, PageRank (Brin and Page, 1998)). The actual burden of IE, which includes reading documents, filtering relevant facts, and extracting an answer, was entirely offloaded to the user.

Modern search engines extensively integrate IE and knowledge graphs to enhance query results. By extracting entities and relations from the web at scale, search engines can present direct answers and rich snippets, rather than only ten blue links (Balog, 2018). For example, Google's search results leverage the Google Knowledge Graph to show knowledge panels[5] alongside traditional links (Singhal, 2012). Such results ground the search experience in a curated graph of facts, complementing algorithmic ranking with semantic understanding.

Recent advancements in LLMs and semantic search have led to a further paradigm shift in the form of generative IR (Metzler et al., 2021; Najork, 2023). In this new regime, the search engine assumes the role of an extractor and synthesizer, processing multiple documents to generate a direct, natural language answer. This transition is not merely an interface update, but represents a fundamental change in how information is accessed, grounded, and monetized (Zhu et al., 2025).

2.5.2 Knowledge Base Construction and Completion

A central application of IE is constructing and maintaining knowledge graphs at web scale. IE systems continuously extract entities and relations from text, tables, and markup and fuse them with existing KGs (Weikum et al., 2021). Traditional efforts include NELL (Carlson et al., 2010), which learned to read the web and incrementally grow a KB over years. Open-source KBs such as DBpedia (Auer et al., 2007) and Wikidata (Vrandecic, 2012) distill Wikipedia into structured triples for wider application (Hogan et al., 2021).

Once an initial KB is constructed via IE, KB completion techniques infers missing facts to achieve high recall. These methods use link-prediction or rule induction to further complete the

[5] https://support.google.com/knowledgepanel/answer/9163198?hl=en

graphs (Bordes et al., 2013; Socher et al., 2013; Yang et al., 2015), leverage Graph Neural Networks (GNNs) to capture higher-order neighborhoods for more accurate inference (Schlichtkrull et al., 2018). Recently, LLMs have been used for semi-automated ontology construction and zero-shot fact checking, reducing the cost of domain adaptation (Pan et al., 2024). Achieving high coverage and calibrated confidence remains critical, as missing facts often hurt downstream applications more than tolerating limited noise.

2.5.3 Question Answering

IE has been a backbone for open-domain question answering systems (Lan et al., 2021). Traditional QA pipelines employ retrieval-based architectures: given a factoid question, the system first uses an IR to fetch relevant documents or passages from a large text corpus (e.g. Wikipedia, search engine), and then a reading module extracts or generates the answer from those texts (Chen et al., 2017). While retriever-reader type QA systems achieve precise extraction, they are limited by explicit answers mentioned as contiguous spans in text (Rajpurkar et al., 2016; Luo et al., 2022). In recent years, dense retrievers and retrieval-augmented generation (RAG) with LLMs overcome this by synthesizing answers (Karpukhin et al., 2020; Lewis et al., 2020b). Moreover, IE and IR techniques help in aggregating evidence across heterogeneous sources for accurate answer generation (Sun et al., 2019; Izacard and Grave, 2021; Christmann et al., 2023; Talmor and Berant, 2018).

A specialized challenge in QA is temporal QA, where correct answers depend on understanding time-based constraints. Work by Jia et al. (2018) decomposes complex temporal questions into simpler sub-queries that can be answered over a knowledge graph with temporal information, and then recombines the results to produce a final answer. Moreover, datasets such as TimeQA (Chen et al., 2021) focusing on questions about time-evolving facts, TORQUE (Ning et al., 2020) tackling temporal ordering questions in text, and CronQuestions (Saxena et al., 2021) covering questions over temporal knowledge graphs have been introduced to benchmark recent models (Mavromatis et al., 2022). Overall, temporal IE extracts events, their timestamps, and temporal relations between them, supporting QA systems (see Section 2.1.1.4).

3

Information Extraction from the Web

3.1 Introduction

3.1.1 Motivation and Background

This chapter focuses on information extraction (IE) from web-scale documents. Specifically, the aim is to perform relation extraction (RE), the core component of IE, from text documents. RE is an important natural language processing (NLP) task with a range of downstream applications (Han et al., 2020; Zhao et al., 2024b; Qin et al., 2024). For reliable usage, it is vital to understand the quality of RE results. While existing extractors provide confidence (or precision) scores, we bring forward the notion of *document coverage* (or recall). Given an input document and an RE method, coverage measures the fraction of the extracted relational tuples compared to the complete ground truth that holds in reality. This notion is considered on a per-subject and per-predicate basis, for example, "*all organizational memberships of Bill Gates*" or "*all companies founded by Elon Musk*".

Document coverage for RE varies highly. Consider the three text snippets about Tesla Inc. shown in Figure 3.1. The first text mentions all five founders of Tesla, while the second text contains only two of them, and the third has just one. In other words, for the entity Tesla and the relation `founded-by`, the first text has coverage 1, the second text has coverage 0.4, and the third text has coverage 0.2. So, when running a reliable RE system over these documents in isolation, the final yield, or output, in terms of recall corresponds to the respective coverage scores.

When applying RE at scale, for example, to populate or augment a knowledge base (KB), an RE system may need to process a large number of input documents that differ widely in coverage. As state-of-the-art extractors are based on heavy-duty neural networks (Lin et al., 2016; Zhang et al., 2017; Baldini Soares et al., 2019; Yao et al., 2019), processing all documents in a large corpus may be expensive, if not prohibitive. Instead, prioritizing the input documents by identifying the best documents with high coverage could be more effective. For instance, processing only the first document shown in Figure 3.1 would result in accurate and complete extraction for "*all founders of Tesla*." This is why *coverage prediction* is crucial for large-scale RE.

The problem would be simple if we could first run an RE system on each document and then assess the yield, either by comparison to withheld labeled data or by sampling followed by human inspection. However, this is exactly the computational bottleneck one must avoid. The challenge is to estimate document coverage, for a given entity and relation of interest, with inexpensive and

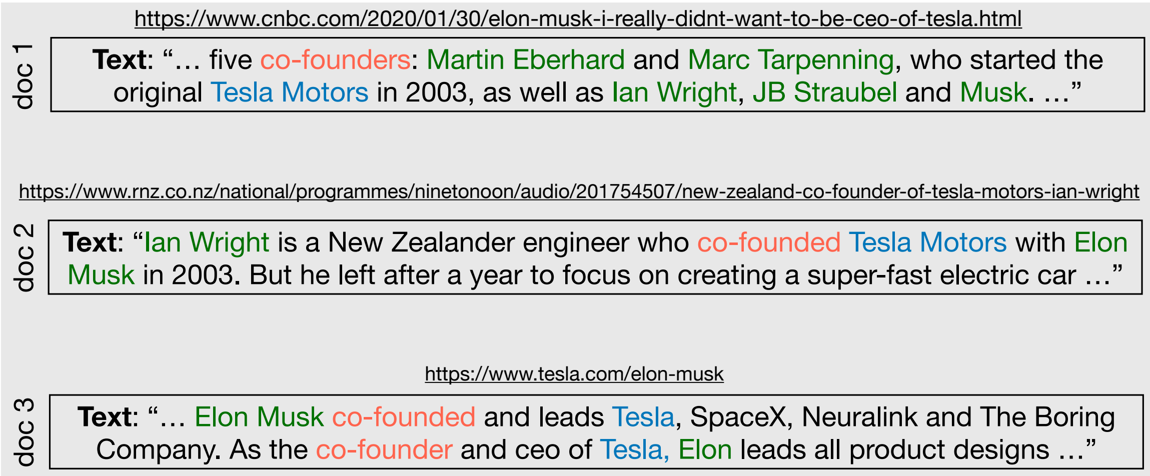

Figure 3.1: Sample documents from the Document Coverage (DoCo) dataset.

lightweight techniques for document processing, and then run the RE system for downstream application.

3.1.2 Research Questions

This goal entails the following research questions:

RQ1: How can we efficiently predict document coverage, and which documents features contribute most to effective prediction? (Section 3.3)

RQ2: How robust is coverage prediction across different entity and relation types? (Section 3.6)

RQ3: How does coverage-based document prioritization help maximize RE recall under resource constraints? (Section 3.8)

3.1.3 Approach

We present the first systematic approach for analyzing and predicting *document coverage for relation extraction*. A novel classifier architecture, named Herb (for Heuristics with BERT), is designed to predict document coverage. Herb efficiently combines lightweight document features such as document length, entity saliency and frequency, website popularity, text complexity, and predicate-related cues in the text. Pretrained language models like BERT (Devlin et al., 2019) are also incorporated into Herb without any costly re-training or fine-tuning for effective prediction.

To facilitate an extensive experimental study on this novel task, we construct a large-scale labeled dataset termed DoCo (for Document Coverage). DoCo consists of 31,366 web documents associated with 520 distinct entities spanning 8 relations, provided as input, along with corresponding automated extractions and coverage labels as output. Tables 3.1 and 3.2 show representative samples of entity-relation-document triples for two types of subject entities, person and organization, from our DoCo dataset.

The best configuration of Herb achieves macro-averaged F1-score of 46%. The classifier provides scores for its predictions and thus also supports ranking documents by their expected yield for the

RE task at hand. We evaluate our approach against a range of state-of-the-art baselines. Our results show that document features alone have only moderate predictive power. However, in combination with pretrained language models, the classifier gives useful predictions of document coverage. Finally, we study the role of coverage prediction in three extrinsic use cases: *KB construction*, *budget-constrained RE*, and *claim refutation*.

1. For KB construction, we show that coverage estimates by Herb are effective in ranking candidate documents and can substantially reduce the number of web pages that need to be processed to build a reasonably complete KB.

2. For budget-constrained RE, we show that prioritizing documents using coverage estimates from Herb as a preprocessing step, before running a full-fledged state-of-the-art RE system, leads to up to 1.63× higher yield under fixed time budget constraints.

3. For claim refutation (e.g., refuting extractions like "Tim Cook is the CEO of Microsoft"), we show that coverage estimates for different documents can provide counter-evidence that can help to invalidate false statements obtained by RE systems.

Overall, these evaluation studies highlight the importance of coverage prediction as a practical and scalable strategy for optimizing document processing in large-scale RE applications.

3.1.4 Contributions

The salient contributions of this work are:

1. We introduce the novel task of predicting document information coverage for RE with a comprehensive analysis of various baselines and features towards its prediction.

2. To support experimental comparisons, we present a large dataset of annotated web documents.

3. We devise lightweight feature-based methods for coverage estimation and analyze their effectiveness both in isolation and when combined with an inexpensive embedding-based document model.

4. We study the application of the classifier on three important use cases: KB construction, resource-bounded relation extraction, and claim refutation. Experiments show that the designed predictor is useful in all of these tasks.

Our data, models and code is available at https://www.mpi-inf.mpg.de/document-coverage-prediction.

3.2 Problem Definition

We take an entity-centric perspective and view RE methods as functions mapping document-entity-relation triples onto the set of objects found in the document. Formally, given a document d, an entity e, a relation r, a ground truth $\texttt{GT}$ of objects that stand in relation r with e, and a relation extraction method $\texttt{extr}$, the document coverage of d for (e, r) applying $\texttt{extr}$ is defined as:

$$\text{coverage}_{\texttt{extr}}(d, e, r) = \frac{|\ \texttt{extr}(d, e, r) \cap GT\ |}{|\ GT\ |} \tag{3.1}$$

For illustration, consider the text snippets shown in Figure 3.1. Here, e is Tesla, r is `founded-by`, and `GT` comprises five triples: (`Tesla`, `founded-by`, `Martin Eberhard`), (`Tesla`, `founded-by`, `Marc Tarpenning`), (`Tesla`, `founded-by`, `Ian Wright`), (`Tesla`, `founded-by`, `Jeffrey Brian Straubel`), and (`Tesla`, `founded-by`, `Elon Musk`). Assuming `extr` is an accurate relation extractor, the coverage for the three documents would be 1, 0.4, and 0.2 respectively, based on their overlap with the ground-truth triples. To automate this process, the task thus takes the form of a prediction task, where the goal is to accurately estimate document coverage given a document.

3.3 Methodology

We propose a set of feature-based, statistical, and neural methods that can be used to predict coverage for a given document. The goal is to estimate coverage by processing unstructured text through inexpensive, lightweight techniques. This is crucial for identifying promising documents before embarking on heavy-duty relation extraction (RE) models. Furthermore, based on empirical observations, we further devise hybrid methods that combine best-performing techniques, while again maintaining low computational cost.

3.3.1 Feature-based Methods

We devise a suite of document-level features that can potentially indicate document coverage.These features are intuitive, easy to compute, and serve as strong signals for downstream modeling.

Document Length. The length of a document is a proxy for the amount of information contained. Longer documents may express more relations and thus have higher potential coverage.

NER Frequency. Length alone can be misleading when a document is verbose yet uninformative. We therefore consider the count of named-entity mentions matching the relation domain (e.g., persons for the relation `family`, or organizations for the relation `member-of`), which could correlate with coverage.

Entity Saliency. The frequency of entity mentions is another important signal. Documents where the entity appears more often are more likely to express relations involving that entity.

IR-Relevance Signal. The surface similarity of the entire document with the input query offers another useful cue. We adopt BM25 (Robertson and Zaragoza, 2009), a classical and powerful information retrieval (IR) model, for ranking documents using $\langle e \rangle + \langle r \rangle$ as the query, where e and r are the target entity and relation, respectively. Moreover, neural (re)-rankers are considered as well (Nogueira and Cho, 2019). We follow Nogueira et al. (2020) and use the T5 sequence-to-sequence model (Raffel et al., 2020) to rank documents and better capture contextual relevance.

Website Popularity. Popular websites may be visited often because they tend to host information-rich content. We use the standard Alexa rank[1] as a measure of website popularity.

[1] https://kinsta.com/blog/alexa-rank/

Entity	Relation	Text Snippet	Coverage
George W. Bush	`family`	President Bush grew up in Midland, Texas, as the eldest son of **Barbara and George H.W. Bush** ... and met **Laura Welch**. They were married in 1977 ... twin daughters: **Barbara**, married to **Craig Coyne**, and **Jenna**, married to **Henry Hager**. The Bushes also are the proud grandparents of **Margaret Laura "Mila"**, **Poppy Louise**, and **Henry Harold "Hal" Hager**; and **Cora Georgia** and **Edward Finn Coyne** ... family also includes **Bob** the cat and **Freddy** the dog ...	1
Warren Buffett	`member-of`	He formed **Buffett Partnership Ltd.** in 1956, and by 1965 he had assumed control of **Berkshire Hathaway** ... Following Berkshire Hathaway's significant investment in **Coca-Cola**, Buffett became ... director of **Citigroup Global Markets Holdings**, **Graham Holdings Company** and **The Gillette Company**. In June 2006 Buffett made an announcement that he would be giving his entire fortune away to charity ...	0.8
Indra Nooyi	`edu-at`	Nooyi was born in Chennai, India, and moved to the US in 1978 when she entered the **Yale School of Management** ... secured her B.S. from **Madras Christian College** and her M.B.A. from **Indian Institute of Management Calcutta** two of India's most prestigious universities ...	0.75
J. K. Rowling	`profession`	Rowling is one of the best-selling **authors** today ... she moved to London and took up the job of a **researcher** and bilingual **secretary** for Amnesty International ... position of a **teacher** led to her relocating to Portugal, wherein she spent the night teaching English ...	0.67

Table 3.1: Samples of entity-relation-document triples for all per-type relations in our `DoCo` dataset.

Entity	Relation	Text Snippet	Coverage
FedEx	`partner-org`	FedEx Corp. ... to acquire **ShopRunner**, the e-commerce ... acquires the International Express business of **Flying Cargo Group** ... acquires **Manton Air-Sea Pty Ltd**, a leading provider ... acquires **P2P Mailing Limited**, a leading ... acquires **Northwest Research**, a leader in inventory ... acquires **TNT Express** ... acquires **GENCO** ... acquires **Bongo International** ... acquires the **Supaswift businesses** in South Africa ... acquires **Rapidão Cometa**, one of the largest transportation ...	1
Apple Inc.	`founded-by`	**Steve Jobs**, the co-founder of Apple Computers, had been trying to hire Pepsi's John Sculley since early 1983 ... switched over to managing the Apple "Macintosh" project that was started by Jef Raskin. Jobs was determined that the new "Macintosh" was going to have a graphical user interface ...	0.33
Intel	`board-member`	**Andy D. Bryant** stepped down as chairman and the board elected lead independent director **Dr. Omar Ishrak** to succeed Bryant as an independent chairman, effective immediately. Intel also announced that **Alyssa Henry** was elected to Intel's board. Her election marks the seventh new independent director added to Intel's board since the beginning of 2016 ...	0.125
3M	`ceo`	The American multinational conglomerate corporation 3M was formerly known as Minnesota Mining and Manufacturing Company. It's based in the suburbs of St.Paul in Maplewood of Minnesota ... 3M has over 88,000 employees all over the world and provides over 55,000 products all over ranging ...	0

Table 3.2: Samples of entity-relation-document triples for all org-type relations in our `DoCo` dataset.

Text Complexity. The complexity of text influences how effectively RE systems can extract relational information and documents written in simpler language often yield higher extraction quality (Han et al., 2020). We use the Flesch score (Flesch and Gould, 1949), a popular text readability measure to quantify text complexity.

3.3.2 Statistical and Neural Methods

We devise several inexpensive statistical and neural models for document representation. We then feed these representations into a logistic regression classifier for coverage prediction.

Latent Topic Modeling (LDA). Latent topics within a document can serve as useful indicators of coverage. For example, for the relation `family`, topics such as *ancestry* or *personal life* are particularly relevant. We use Latent Dirichlet Allocation (Blei et al., 2003) to model each document as a distribution over latent topics.

BOW with TFIDF. A simple yet effective statistic to measure word importance given a document in a corpus is the product of term frequency and inverse document frequency (TF-IDF). We vectorize each document into a Bag-of-Words (BOW) representation with TF-IDF weighting to capture lexical salience.

N-grams with TFIDF. To capture contextual patterns, we extend the BOW representation to include frequent n-grams ($n \leq 3$) with TF-IDF weights. This allows the model to encode short expressions relevant to relation mentions.

LSTM. Previous work by Razniewski et al. (2019) used textual features to estimate the presence of a complete set of objects in a text segment. We adopt their architecture, representing documents using 100 dimensional GloVe embeddings (Pennington et al., 2014), and processing them in LSTM (Hochreiter and Schmidhuber, 1997), followed by a feed-forward layer with ReLU activation before the classifier.

Language Model (BERT). Without costly retraining or fine-tuning, we adopt a feature-based approach using pretrained BERT embeddings (Devlin et al., 2019). We extract activations from the last four hidden layers, aggregate them, and feed the resulting contextual representations into a two-layer, 768-dimensional BiLSTM before classification. This setup efficiently leverages BERT's linguistic capabilities while keeping computation lightweight.

3.3.3 Hybrid Methods

Our experiments (in Section 3.6) reveal that each of the proposed document features (in Section 3.3.1) possesses only moderate predictive power when used in isolation. We therefore formulate a lightweight classifier to combine document features with the best-performing statistical model (TF-IDF), or language model (BERT). These hybrid methods serve as a middle ground between purely interpretable feature-driven and fully neural systems, making them particularly suitable for large-scale or resource-constrained RE pipelines.

Heuristics with BOW+TFIDF (Heu+TFIDF). We combine TF-IDF features with all the six document features using a stacked Logistic Regression (LR) framework, as shown in Figure 3.2. In

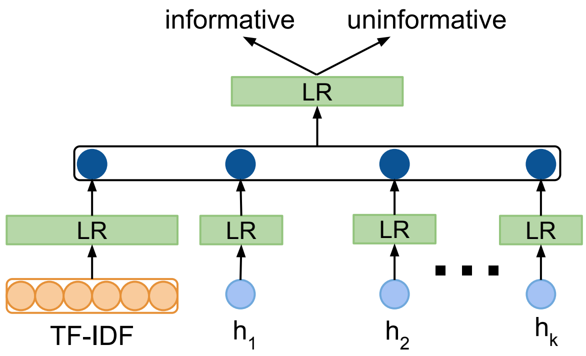

Figure 3.2: Architecture for features combined with TF-IDF (Heu+TFIDF)

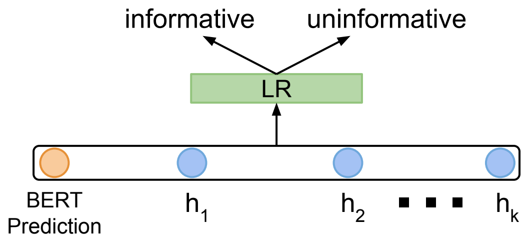

Figure 3.3: Architecture for features combined with BERT Prediction (HERB)

the first level, the TF-IDF vector and each individual feature are fed into separate LR classifiers to produce preliminary predictions. In the second level, all the outputs from the previous level are concatenated and fed into a final LR classifier for the final coverage prediction. The entire model is trained jointly in a supervised fashion.

Heuristics with BERT (HERB). We combine BERT representations with all the six document features in a two-step process, as shown in Figure 3.3. In the first step, we reuse the pretrained BERT model described earlier in Section 3.3.2, without any additional training or fine-tuning, to obtain an initial coverage prediction. In the second step, the BERT-based prediction is concatenated with all the document features to form a single feature vector, which is then fed into a Logistic Regression classifier. This allows HERB to integrate semantic information captured by BERT with interpretable and low cost feature signals, leading to efficient yet strong performance across relations.

3.4 Dataset for Evaluation

A thorough study of document coverage prediction requires a corpus with two characteristics: (i) relation diversity, i.e., documents containing enough automatically extractable relations, and (ii) content diversity, i.e., multiple documents with varying content per entity. Existing text corpora, like the popular NYT (Sandhaus, 2008) and Newsroom dataset (Grusky et al., 2018), contain ample numbers of articles that mention newsworthy entities; however, the articles are primarily short, mentioning only very few relations. On the other end, machine-translated multilingual versions of Wikipedia articles (Roy et al., 2020) allow extraction of many relations but lack diversity.

For the novel task of predicting document information coverage, we thus built the `DoCo` (Document

Coverage) dataset, consisting of 31,366 web documents for 520 distinct entities, each with its coverage value. Figure 3.4 illustrates the dataset construction.

Entity Selection. First, well-known entities of two types, person (PER) and organization (ORG), were selected from popular ranking lists by Time 100[2] and Forbes[3][4] ("Influential people around the globe", "Most valuable tech companies"). These entities covered 12 diverse sub-domains, including politicians, entrepreneurs, singers, sportsmen, writers, actors, for PER, and technology, automobile, retail, conglomerate, pharmaceuticals, financial corporations, for ORG. Popular and long-tail entities for PER, companies across demographics and with differing net worth for ORG, were chosen to further obtain documents with varying content.

Websites & Content. We aimed to collect diverse 100 URLs per entity by issuing a set of search engine queries per entity, e.g., "about PER", "PER biography", "ORG history". A total of 6 set of queries for PER and 10 for ORG was designed. Since the URLs returned over the set of queries were not always unique, we retained the duplicated URL only once.

Extracting textual content without noisy headers, menus, and comments, required a labor-intensive scraping step. We handled the multi-domain content scraping task through a combination of libraries like Newspaper3k[5], Readability[6], and online scraping services like Import.io[7] and ParseHub[8]. We ensured high-quality scraped content by applying rule-based filters to remove noisy elements like embedded ADs and reference links. The scraped documents covered a range of website domains, including biographical sites, news articles, official company profiles, newsletters, and so on.

Relation Tuples. Each document in `DoCo` was processed by two relation extraction APIs, Rosette[9] and Diffbot[10]. To annotate each document with coverage, we focused only on the entity queried initially to obtain the document. For our experimental study, we selected the following frequently occurring relations: `member-of`, `family`, `edu-at`, and `profession`, for PER, and `partner-org`, `founded-by`, `ceo`, and `board-member`, for ORG. For more accurate coverage calculation, the RE tuples were deduplicated, e.g., (Gates, `member-of`, Microsoft Corp.) would become (Bill Gates, `member-of`, Microsoft), via alignment to Wikidata identifiers returned by the APIs.

The relations extracted by the APIs are fine-grained like person-member-of, person-employee-of, org-acquired-by, and org-subsidiary-of. We combined the first two as member-of for PER and the last two as partner-org for ORG as coarse-grained relations.

Ground Truth. We considered three ground-truth formulations to calculate coverage for each document:

[2] https://time.com/collection/100-most-influential-people-2020/
[3] https://forbes.com/forbes-400/
[4] https://forbes.com/lists/global2000/#9a993675ac04
[5] https://newspaper.readthedocs.io/en/latest/
[6] https://pypi.org/project/readability-lxml/
[7] https://www.import.io/
[8] https://www.parsehub.com/
[9] https://rosette.com/
[10] https://www.diffbot.com/

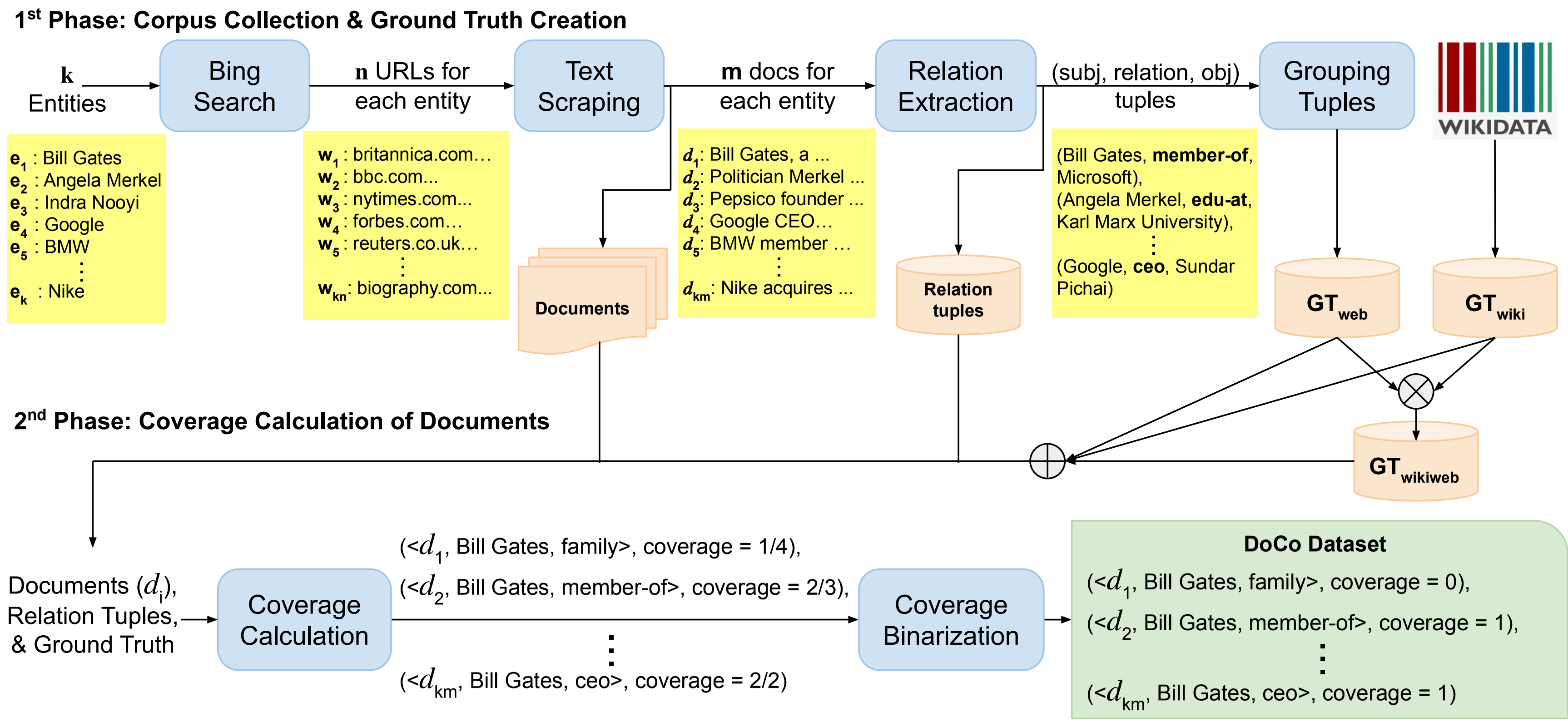

Figure 3.4: **Dataset Construction Pipeline.** There are two main phases: 1) corpus collection to create $\texttt{GT}_{\texttt{web}}$, and 2) coverage calculation. Phase 1 involves: i) for each entity e_i, n websites are collected using the Bing search API, ii) text is scraped from each website, iii) RE tuples from documents are extracted via Rosette/Diffbot, and iv) RE tuples are deduplicated and consolidated to form $\texttt{GT}_{\texttt{web}}$. The scraped documents are stored as inputs for phase 2 which consists of: i) for each document d_i, previously extracted relations are collected, and ii) based on the choice of GT, coverage is calculated to create the final DoCo dataset.

Relation	Wikidata Property
`member-of`	member of (P463), member of political party (P102), part of (P361), employer (P108), owner of (P1830), record label (P264), member of sports team (P54)
`family`	father (P22), mother (P25), spouse (P26), child (P40), stepparent (P3448), sibling (P3373)
`edu-at`	educated at (P69)
`profession`	position held (P39), occupation (P106)
`partner-org`	owner of (P1830), owned by (P127), member of (P463), parent organization (P749), subsidiary (P355)
`founded-by`	founded by (P112)
`ceo`	chief executive officer (P169)
`board-member`	board member (P3320)

Table 3.3: Wikidata property names and identifiers used to create $\texttt{GT}_{\texttt{wiki}}$

1. *Wikidata* ($\texttt{GT}_{\texttt{wiki}}$): A popular KB providing data for most relations yet having coverage limitations (Galárraga et al., 2017; Luggen et al., 2019). For example, for "Bill Gates", "Microsoft" and other popularly associated companies contain details for the `member-of` relation, but niche entities like "Honeywell" have missing entries. Depending on the entity type and sub-domain, we created the ground-truth labels by choosing those Wikidata properties that best matched the semantics of the 8 selected relations. Table 3.3 provides the complete information.

2. *Web Extractions* ($\texttt{GT}_{\texttt{web}}$): We used the set of frequent extractions across all the documents in `DoCo` as web-aggregated ground truth. For a given entity-relation (e, r), an extraction was determined frequent if it appeared in at least 5% of total documents corresponding to e, or if its count was no less than 5 times the highest counted tuple for (e, r). Deciding frequent extractions relative to total document count and other tuples' frequencies for an entity resulted in noise-free ground-truth labels.

3. *Wikidata and Web Extractions* ($\texttt{GT}_{\texttt{wikiweb}}$): We merged both previous variants using set union operation and phrase embeddings with cosine similarity for higher recall.

Coverage Calculation. Coverage was computed on a per entity-relation-document basis using equation 3.1. Even though real-valued coverage values are computed while constructing the dataset, it is often not possible to give nuanced predictions at test time. Consider the text "... *Musk is a co-founder of Tesla ...*". The term `co-founder` clearly indicates the presence of multiple founders; however, the context does not provide any clue on the total number of co-founders. For example, there could be one other co-founder (coverage 0.5) or 9 other co-founders (coverage 0.1).

Coverage Binarization. We binarized the coverage values to circumvent the above problem, splitting documents into two classes: *informative* and *uninformative*. The binarization method comprised

# PER entities	250
# ORG entities	270
# Relations	8
# Documents	31,366
Doc. length range (words)	[20, 10906]
# Unique website domains	600
# Doc. with non-zero RE tuples	26956
# Doc. with non-zero coverage	14086
# Doc. in class informative	7103 (22.6 %)

Table 3.4: Characteristics of the `DoCo` dataset

Relation	$\texttt{GT}_{\texttt{wiki}}$	$\texttt{GT}_{\texttt{web}}$	$\texttt{GT}_{\texttt{wikiweb}}$
`member-of`	3.61	6.51	7.12
`family`	2.21	4.0	4.41
`edu-at`	2.26	2.07	2.58
`profession`	5.86	7.76	10.37
`partner-org`	6.16	4.26	3.12
`founded-by`	1.07	1.06	1.66
`ceo`	1.03	2.77	2.86
`board-member`	0.47	1.44	1.75

Table 3.5: Average number of objects per entity

of an absolute and a relative threshold: a document was labeled as informative or 1 if its coverage was greater than 0.5, or greater than the coverage of at least 85% of documents for the same (e, r); otherwise, it was labeled as uninformative or 0.

Dataset Characteristics. After filtering duplicates, irrelevant URLs like social media handles, and video-content websites, we obtained a total of 31,366 documents for 520 entities. Table 3.4 provides an overview of the `DoCo` dataset. We can see that `DoCo`'s labels are imbalanced, as only 22.6% of the documents are informative and 77.4% are uninformative. The count of documents with non-zero RE tuples is higher than those with non-zero coverage since the RE tuples were not always related to the subject entity, hence irrelevant towards coverage calculation.

Table 3.5 gives the average number of objects present in each ground truth variant. On average across relations, the number of objects in $\texttt{GT}_{\texttt{web}}$ is higher than those in $\texttt{GT}_{\texttt{wiki}}$ by 23.7%, and $\texttt{GT}_{\texttt{wikiweb}}$ is higher than those in $\texttt{GT}_{\texttt{wiki}}$ by 28.8%. This implies that $\texttt{GT}_{\texttt{web}}$ and $\texttt{GT}_{\texttt{wiki}}$ can have overlapping objects, and $\texttt{GT}_{\texttt{web}}$ might contain extra objects towards $\texttt{GT}_{\texttt{wikiweb}}$ creation.

Dataset Quality. We analyzed the quality of the `DoCo` dataset by comparing automatic relation extractions to extractions given by human annotators. A sample of 400 documents was selected, 50 per relation, with half from the high-coverage range and the rest from the low-coverage range.

Relation	Human	Diffbot	Rosette	GT_{wiki}	GT_{web}	$GT_{wikiweb}$
member-of	4.36	3.66	**5.04**	4.54	7.12	9.22
family	4.74	**3.82**	0.66	1.76	5.78	6.64
edu-at	1.72	2.5	**2.52**	2.94	3.08	2.18
profession	2.9	**4.26**	-	6.7	6.14	9.52
partner-org	3.7	0.72	**2.26**	0.8	5.04	5.92
founded-by	1.34	0.58	**1.8**	0.78	2.84	2.96
ceo	2.02	**1.96**	-	1.68	4.32	4.2
board-member	2.62	**1.54**	-	2.82	3.48	2.64

Table 3.6: Average tuple count per relation. The RE tool with higher tuple count (boldfaced) is chosen for each relation.

Each document was annotated with all correct tuples for the document's main subject entity.

Table 3.6 shows the observed averaged counts. We note that the human annotators extracted a substantial number of tuples for all 8 relations, indicating the richness and breadth of the DoCo documents. The two automatic extractors mostly yielded smaller numbers of tuples, with a few exceptions. These exceptions include spurious tuples, though. The ground-truth variants consistently suggest higher numbers, but except for the conservative GT_{wiki}, these are usually overestimates due to spurious tuples. The GT variants should thus be seen as upper bounds for the true RE coverage.

We analyzed how well the automatic annotations reflect human annotations' coverage by computing Pearson correlation coefficients for the entire set of 400 sample documents. For a relation, the RE tool with higher averaged count was chosen for our experiments, and the correlation for (Human, RE) is 0.68. This shows that optimizing for coverage by automatic RE tools is highly correlated with the overarching goal of approximating human-quality outputs.

3.5 Experimental Setup

Dataset. We considered two automatic RE tools, Rosette and Diffbot, as extr, and three ground truth variants: GT_{wiki}, GT_{web}, $GT_{wikiweb}$. For each relation, we report on the combination of RE tool and GT variant that achieves the highest count of documents classified as high-coverage.

Each relation had a separate labeled set of documents, split into 70% train, 10% validation and 20% test. Information leakage was prevented by splitting along entities, i.e., all documents on the same entity would exclusively be in one of train, validation or test set. The number of training samples per relation varies from 664 (board-member) to 3604 (profession). Since the label distribution in DoCo is imbalanced, the uninformative (or 0) class in all train datasets were undersampled to obtain a 50:50 distribution, while the validation and test datasets were kept unchanged to reflect the real-world imbalance. Named entities and numbers were masked.

Methods. Each proposed document feature (as detailed in Section 3.3.1) was turned into a classifier by first ranking documents according to the feature, and then labeling the top 50% documents as class 1 or informative. We used the Okapi BM25[11] and monoT5[12] open-source implementations for IR ranking. The monoT5 model is generally used for passage ranking, and as `DoCo` documents are much longer with multiple passages, we used the MaxP algorithm (Dai and Callan, 2019) to compute the document ranking. Since the difference in performance between T5 and BM25 models is negligible, we chose the simpler yet equally effective BM25 model as IR-relevance signal for Herb. We also contrast the predictive power of our proposed methods with two random baselines: a fair coin, and a biased coin maintaining the label imbalance in our test set. These random baselines predict a coverage estimate between $[0, 1]$.

Feature based methods including topic modeling with LDA, TF-IDF and N-grams, were fed to a Logistic Regression classifier. In the LSTM architecture, we used 100 dimensional GloVe embeddings with a vocabulary size of $100,000$, and a 100 dimensional hidden state for LSTM. For methods involving pretrained language models, we used the BERT-base-uncased[13], without additional retraining or fine-tuning, to encode sentences, by summing the [CLS] token's representation from the last four hidden layers. Input documents were padded or truncated to 650 sentences, and represented through sentence encodings. Coverage classification was performed using the feature-based approach outlined in Devlin et al. (2019).

We constructed mini-batches of size 32, used the Adam optimizer initialized with a constant learning rate of $1e$-05 and $1e$-09 epsilon value, and trained for 200 epochs. Since our dataset is imbalanced, we monitored validation precision to save the best model, and report optimal F1-scores (Lipton et al., 2014) to compare results.

3.6 Results

Our results are shown in Tables 3.7 and 3.8. Each feature-based model gives a mediocre performance, with T5 IR achieving the highest average F1 of 23.6 among the feature-based methods. In the trained group of models, LDA has the lowest average F1 of 16.9, while BERT performs the best with an average F1 of 36.2. Although each feature-based model has moderate predictive power, combining them with statistical models like TF-IDF, or pretrained BERT model, gives the best performance. Among the combination models, Herb outperforms Heu+TFIDF in a clear majority of relations.

Model Analysis. Statistical models like BOW+TFIDF and Ngrams+TFIDF performed comparably to BERT for a minority of relations. To better understand these models, we analyzed highly positive and negative features. Table 3.9 provides noteworthy examples. We observe the presence of semantically relevant phrases. We also inspect the weights of the trained LR classifier of Herb. Across relations, BERT had the highest average weight (5.05), followed by BM25 (2.56), while NER Count had the lowest weight (0.07).

Feature Ablations. We further perform an ablation analysis, with Table 3.10 showing the average F1-scores when individual document features are removed from Herb. Removing either BM25 or

[11]https://pypi.org/project/rank-bm25/
[12]https://github.com/castorini/pygaggle
[13]https://huggingface.co/bert-base-uncased

Method	`member-of`	`family`	`edu-at`	`profession`	Average
Random (biased)	5.7	6.8	4.9	10.0	6.9
Random (fair)	15.7	11.1	12.6	15.4	13.7
Text Complexity	9.6	5.4	6.1	10.3	7.9
Alexa Ranking	12.6	9.8	8.1	12.4	10.7
Entity Saliency	17.8	14.3	11.9	18.2	15.6
Document Length	20.5	19.0	15.5	21.9	19.2
NER Count	24.3	19.8	18.2	-	20.8
BM25 IR	27.1	21.1	18.8	26.3	23.3
T5 IR	26.9	23.2	20.3	29.6	25.0
LDA Topic Model	19.3	19.0	14.5	21.1	18.5
GloVe+LSTM	16.5	28.6	19.8	32.9	24.5
Ngrams+TFIDF	36.2	40.0	25.6	40.2	35.5
BOW+TFIDF	36.0	41.0	29.2	42.1	37.1
BERT	40.4	39.7	35.7	44.4	40.1
Heu+TFIDF	41.9	**43.5**	31.3	36.5	38.3
HERB	**44.2**	41.7	**40.5**	**45.6**	**43.0**

Table 3.7: F1-scores (%) obtained on the coverage prediction task for per-type relations by various methods.

Text Complexity leads to a significant drop in performance, indicating that other features or BERT alone do not capture the task specific signals well.

Human Performance. Finally, we compare the results against human performance on identifying high-coverage documents. For each relation, 10 randomly sampled test documents were labeled as informative or uninformative for RE solely by reading the document. Averaged over all relations, humans obtained an F1 score of 70.42%, compared to HERB predictions reaching an average F1 of 39.3%, and all baselines were significantly inferior. The large gap between humans and learned predictors shows the hardness of the coverage prediction task and underlines the need for the presented research.

3.7 Analysis and Discussion

Domain Dependency. To investigate how strongly prediction depends on in-domain training data, we performed a stress test, where the train, validation and test set were split along domains (e.g., singers vs. entrepreneurs vs. politicians). Tables 3.11 and 3.12 show the resulting F1-scores (%). For HERB, the average F1-score on the in-domain test set is 34.3%, while on the out-of-domain test set is 34.2%, i.e., there is no notable drop for the challenging domain-transfer case. We observe a minor drop for larger relations, while even increases are visible for the smallest two relations. This suggests that HERB learned generalizable features that are beneficial across domains.

Method	`partner-org`	`founded-by`	`ceo`	`board-member`	Average
Random (biased)	7.5	1.2	13.5	3.7	6.5
Random (fair)	15.2	8.9	21.3	7.2	13.2
Text Complexity	3.5	3.3	15	5.4	6.8
Alexa Ranking	16.7	11.3	24.8	7.3	15.0
Entity Saliency	14.7	8.4	24.6	7.1	13.7
Document Length	23.9	12.8	28.8	8.5	18.5
NER Count	21.1	13.7	34.5	11.8	20.3
BM25 IR	21.8	12.9	36.6	12.1	20.9
T5 IR	19.5	15.4	41.1	13.1	22.3
LDA Topic Model	15.7	8.6	25.2	11.5	15.3
GloVe+LSTM	24.2	19.5	24.4	4.9	18.3
Ngrams+TFIDF	18.6	25.5	41.8	30.2	29.0
BOW+TFIDF	17.2	28.3	40.6	32.1	29.6
BERT	22.0	30.8	43.0	33.8	32.4
Heu+TFIDF	**35.1**	28.2	41.4	22.0	31.7
HERB	28.8	**32.5**	**46.2**	**34.8**	**35.6**

Table 3.8: F1-scores (%) obtained on the coverage prediction task for org-type relations by various methods.

Relation	**Important Phrases**
`member-of`	[org], is part of, ambassador, is associated with, [org] partner
`family`	[person], married, father, wife, children, daughter, parents, [number]
`edu-at`	[org], graduated, degree, studied, [org] in [number], is part of
`profession`	[person], leader, president, actor, professor, writer, founder, police, portman
`partner-org`	[org], [number] [org], subsidiary, merger, the company, member of
`founded-by`	[person], founder, director, executive, chairman, co founder, head of, chief executive
`ceo`	ceo, [person] director, chief, officer, founders, chief executive officer, president
`board-member`	[org], [person], chairman, executive, board of directors, [number] senior executive, officer in charge, representative director

Table 3.9: Highly weighted phrases given by the trained LR classifier of Ngrams+TFIDF and BOW+TFIDF.

Evaluation of Document Ranking. So far, we have evaluated our methods on a binary prediction problem. However, use cases frequently require a ranking capability (see also Section 3.8). We additionally evaluate our methods on a ranking task, where documents are ranked by the score of positive predictions. We use the mean Normalized Discounted Cumulative Gain (mean

Model (variant)	F1-score
Herb	39.3%
- Doc. Length	36.8% (-2.44)
- Entity Saliency	36.4% (-2.85)
- Alexa Ranking	36.3% (-3.03)
- NER Count	36.2% (-3.11)
- BM25	36.0% (-3.29)
- Text Complexity	35.7% (-3.62)

Table 3.10: Average F1 performance with feature ablations. Text Complexity and BM25 are most important.

Method	`member-of`	`family`	`edu-at`	`profession`	Average
BOW+TFIDF	33.97	42.05	38	46.76	40.2
BERT	37.08	42.63	39.31	53.88	43.2
Herb (in-domain)	40.8	41.8	34.9	42.8	40.1
Herb (out-of-domain)	35.7	39.7	32.5	39.1	36.8
Training Data Size	2194	1650	1458	2940	-

Table 3.11: F1 comparison for per-type relations on the in-domain and out-of-domain test set.

Method	`partner-org`	`founded-by`	`ceo`	`board-member`	Average
BOW+TFIDF	20.59	15.17	44.44	24.55	26.2
BERT	35.78	27.2	45.79	31.22	35.0
Herb (in-domain)	28.4	17.1	45.4	23.3	28.6
Herb (out-of-domain)	29.4	23.8	42.3	31.1	31.7
Training Data Size	1124	828	2058	608	-

Table 3.12: F1 comparison for org-type relations on the in-domain and out-of-domain test set.

nDCG) (Järvelin and Kekäläinen, 2002) as the evaluation metric. A similar performance trend to the F1 metric is observed among our methods. Herb performs the best with an average nDCG score of 0.45 across relations, while BERT and Heu+TFIDF have 0.44 and 0.43, respectively.

RE Limitations. The performance of RE methods significantly impacts the quality of $\mathtt{GT}_{\mathtt{Web}}$ as well as the RE coverage of documents. Although we used state-of-the-art commercial APIs, these nonetheless struggle on open web documents. To illustrate this, we randomly sampled 40 documents from `DoCo` and compared the count of RE tuples returned by Diffbot/Rosette against the count by a human relation extractor. Diffbot returned 60.6% fewer relational tuples, and Rosette returned 72.3% fewer, suggesting the need for further improvement of RE methods.

Error Analysis. We analyzed the incorrect predictions by Herb and categorized the errors. For each relation, we randomly sampled 10 incorrectly predicted documents, 5 false positives and 5 false negatives. Out of the total 80 samples, 63.75% of documents contained partial information for the chosen relation; on 15% of documents the IE methods failed to extract all the necessary RE tuples; the ground truth for 3.75% of documents had an incomplete set of objects; 3.75% documents had noisy content; and 2.5% documents had incomplete information due to failure of scraping methods on complex website layouts.

Multiple documents in the low-information category had speculative content, e.g., considerations about candidates for a new appointment as a board member or CEO. In other cases, the document would mention the increased count of board members without their names. A few documents also had partial information leading to false positives, e.g., a document mentioning the footballer Sergio Agüero for the `family` relation was incorrectly classified as informative; as it contained a complete family history about another footballer, Diego Maradona (Sergio's father-in-law).

Conversely, documents may contain information relevant to a relation without actual mention of the relation, which leads to false negatives. For example, a document on the LinkedIn Corporation stating *"...Weiner stepped down from LinkedIn ... He named Ryan Roslansky as his replacement."* was labeled uninformative for the `ceo` relation. Although Ryan Roslansky and LinkedIn are related through the `ceo` relation, the implicit statement was not noticed by Herb.

We specifically inspected the IR baselines' performance to understand better why these are mediocre predictors at best. The IR signals about entire documents merely reflect that a document is on the proper topic given by the query entity, but that does not necessarily imply that the document contains `many` relational facts about the target entity. For RE coverage, IR-style document-query relevance is a necessary cue but not a sufficient criterion.

Efficiency and Scalability. We measured the run-time of Herb against a state-of-the-art neural model for document-level RE (DocRED) (Yao et al., 2019). Based on the DocRED leaderboard[14], we selected the currently best open-source method: the Transformer-based Structured Self-Attention Network (SSAN) (Xu et al., 2021a). A sample of 100 documents from `DoCo` was given to both Herb and SSAN and processed as follows. For Herb, features are computed utilizing BERT, followed by coverage prediction. For SSAN, documents first need to be pre-processed to construct the neces-

[14] https://competitions.codalab.org/competitions/20717#results

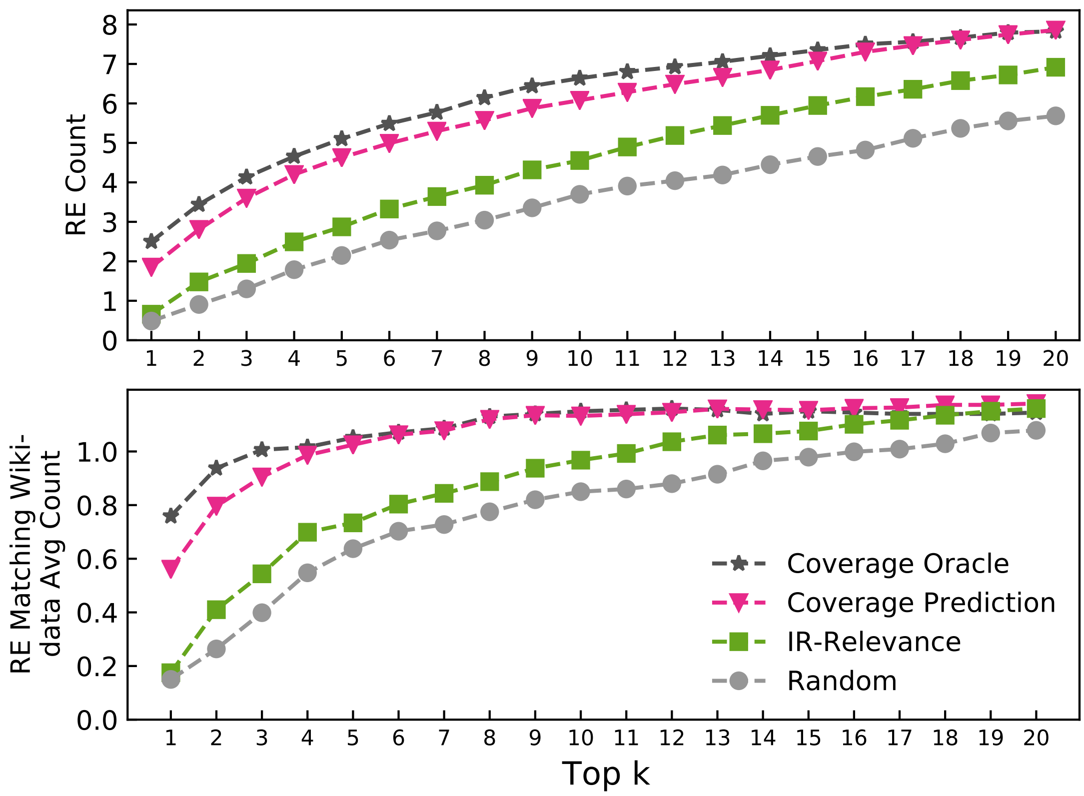

Figure 3.5: Total yield (top) and precision (bottom) of KBC based on different ranking methods for documents.

sary DocRED representation. This includes named entity recognition and pair-wise co-reference resolution, using Stanza[15] to properly group same-entity occurrences.

The measurements show the following. Herb takes ca. 2 seconds, on average, to process one document, whereas SSAN requires 13.6 seconds—a factor of 6.8 higher in speed and resource consumption. The difference becomes even more prominent for very long documents with many named entity mentions. Herb's run-time grows linearly with document length, while SSAN's run-time exhibits quadratic growth with the number of entity mentions. This quadratic complexity of full-fledged neural RE has inherent reasons, as stated in Yao et al. (2019). Document-level relation extraction generally requires computations for all possible pairs of entity mentions. The neural RE methods need to have the positions of candidate entity pairs as input, which necessitates considering all pairs of mentions.

3.8 Extrinsic Evaluation

To demonstrate the importance of coverage prediction, we evaluated its utility in two use cases, knowledge base construction and claim refutation. For the former, we discuss the importance of ranking documents by RE coverage (Section 3.8.1) and a practically relevant setting where RE is

[15] https://stanfordnlp.github.io/stanza/

Method	RE Count	#Docs Processed
SSAN	59	410
HERB+SSAN	96	318

Table 3.13: Relation extraction under run-time constraint.

constrained by resource budgets (Section 3.8.2).

3.8.1 Document Ranking for Relation Extraction

Relation extraction plays a pivotal role in KB construction. We show the relevance of coverage estimates for prioritizing among documents. Entities from our test dataset serve as subjects for RE. We select top k documents from the test dataset corpus by four different techniques. We compare the performance of each method by the total number of extracted RE tuples per subject and compute recall w.r.t. the Wikidata ground-truth.

1. *Random*: A random sample of documents.
2. *IR-Relevance*: Using BM25 to identify the most relevant documents.
3. *Coverage Prediction*: HERB's predictions to rank documents.
4. *Coverage Oracle*: Selecting documents by their ground-truth labels from `DoCo`. This ranking gives an upper bound on what an ideal method could achieve.

Setup. The document coverage calculation is on a per (e, r) pair basis. In a single iteration, all the proposed methods are given a set of documents partitioned by (e, r) pairs. Each method uses its technique to rank the documents, and the top k ranked documents are given to the RE API (Rosette or Diffbot) for obtaining the set of relational tuples.

Results. Figure 3.5 (*top*) compares the total RE tuples obtained by the proposed methods, averaged across test dataset entities and 8 chosen relations. Notably, BM25 doesn't perform much better than random, while coverage prediction is not far behind the perfect ranking defined by the coverage oracle. Ordering documents by coverage prediction instead of IR-relevance gives 50% more extractions from the top-10 documents.

Figure 3.5 (*bottom*) shows the number of RE tuples that match the Wikidata KB, thus comparing the methods on precision. As was foreseeable, the coverage oracle method wins due to the usage of correct coverage values for ranking. HERB's coverage prediction performance is considerably higher than IR-relevance and other methods, while it matches the coverage oracle for $K \geq 4$. Beyond $K > 15$, all methods yield nearly the same sets of tuples, hence similar precision.

3.8.2 Budget-constrained Relation Extraction

Document coverage predictions are particularly important for massive-scale RE tasks targeted at long-tail entities, such as populating or augmenting a domain-specific knowledge base (e.g., about diabetes or jazz music). Such tasks may require screening a huge number of documents. Therefore,

Subject	Relation	Object	Document Snippet
Alphabet Inc.	`ceo`	Susan Wojcicki	Susan Wojcicki is CEO of Alphabet subsidiary YouTube, which has 2 billion monthly users.
Oracle Corporation	`founded-by`	David Agus	Oracle Co-founder Larry Ellison and acclaimed physician and scientist Dr. David Agus formed Sensei Holdings, Inc.
PepsiCo	`board-member`	Joan Crawford	Film actress Joan Crawford, after marrying Pepsi-Cola president Alfred N. Steele became a spokesperson for Pepsi.

Table 3.14: Incorrect claims extracted by Diffbot RE API from documents predicted as low coverage.

practically viable RE methods need to operate under budget constraints, regarding the monetary cost of computational resources (e.g., using and paying for cloud servers) as well as the cost of energy consumption and environmental impact.

In the experiment described here, we simulate this setting, comparing standard RE by SSAN against Herb-enhanced RE where Herb prioritizes documents for RE by SSAN. We assume a budget of 10 minutes of processing time and give both methods 100 candidate documents. SSAN selects documents randomly and processes them until it runs out of time. Herb+SSAN sorts documents by Herb scores for high coverage and then lets SSAN process them in this order. The time for Herb itself is part of the 10-minute budget for the Herb+SSAN method.

As a proof-of-concept, we ran this experiment for a sample of 10 different entities (each with a pool of 100 documents). Table 3.13 shows the results. Due to the upfront cost of Herb, Herb+SSAN processes fewer documents within the 10-minute budget, but its yield is substantially higher than that of SSAN alone, by a factor of 1.63. This demonstrates the need for document-coverage prediction towards realistic usage.

3.8.3 Claim Refutation

Our second use case is fact-checking, specifically the case of refuting false claims by providing counter-evidence via RE.

Reasoning. *Extraction confidence* and *document coverage* are conceptually independent notions. However, when looking at sets of documents, an interesting relation emerges. Consider two documents, d_1 with high coverage, and d_2 with low coverage, along with two claims c_1 and c_2 from the respective documents, extracted with the same confidence. *"Can we use coverage information to make claims about extraction correctness?"*

We propose the following hypothesis: given that d_1 is asserted to have high coverage, we can conclude that any statement not mentioned in d_1 (like c_2) is more likely false. In contrast, the low coverage of d_2 implies that d_2 is unlikely to contain all factual statements. Thus, c_1 not being found in d_2 is no indication that it could not be true.

Validation. We experimentally validated the correctness of the above reasoning as follows. From the collection of relation extractions from the test dataset documents, we randomly sampled 69 pairs of claims for the same entity and relation, which had low support (i.e., extraction found only in one website). We then ordered the pairs by the coverage of the documents that did not express them, obtaining 69 claims with relatively higher coverage in non-expressing documents and 69 claims with relatively lower coverage.

We manually verified the correctness of each claim on the Internet, verifying annotator agreement on a sub-sample, where we found a high Fleiss' Kappa (Fleiss, 1971) inter-annotator agreement of 0.82. Using these annotations, we found that from the 69 claims absent from lower-coverage documents, 58% (40) were correct, while from those absent from higher-coverage documents, only 36% (25) were correct. In other words, the fraction of correct claims absent from low-coverage documents is 1.6 times higher; so coverage can be used as a feature for claim refutation. Table 3.14 shows examples of claims absent from high-coverage documents.

3.9 Related Work

While detailed background on the relevant topics is given in Chapter 2, this section highlights prior work at the time of this project[16].

Relation Extraction (RE). RE is the task of identifying the relation types between two entities that are mentioned together in a sentence or in proximity within a document (e.g., in the same paragraph). RE has a long history in NLP research (Suchanek et al., 2009; Mintz et al., 2009; Riedel et al., 2010), with a overview given by Han et al. (2020). State-of-the-art methods are based on deep neural networks trained via distant supervision (Lin et al., 2016; Zhang et al., 2017; Baldini Soares et al., 2019; Yao et al., 2019). On the practical side, RE is available in several commercial APIs for information extraction from text. In our experiments, we make use of Rosette[17] and Diffbot[18]. Our approach is agnostic to the choice of extractors, where any RE tool can be plugged in.

Knowledge Base Construction (KBC). RE plays a crucial part in the more comprehensive KBC task: identifying instances of entity pairs that stand in a given relation in order to construct a knowledge base (Weikum et al., 2009; Ji et al., 2010; Mitchell et al., 2018; Martinez-Rodriguez et al., 2020; Weikum et al., 2021; Hogan et al., 2021). The input is typically a set of documents, often assumed to be fixed and given upfront. This disregards the critical issue of benefit/cost trade-offs, which mandates identifying high-yield inputs for resource-bounded KBC. Identifying relevant, expressive and preferable sources for KBC is often referred to as *source discovery*. Source discovery can be performed via IR-style ranking of documents or can be based on heuristic estimators of the yield of relation extractors (Wang et al., 2019; Razniewski et al., 2019). The former work, in particular, approaches yield optimization as a set coverage maximization problem through shared

[16] in the year 2022-2023
[17] https://rosette.com/
[18] https://www.diffbot.com/

properties of extracted entities. The latter uses textual features in a supervised SVM or LSTM model, a baseline with which we also compare in our experiments.

Document Ranking in IR. Information retrieval (IR) ranks documents by relevance to a query with keywords or telegraphic phrases. Relevance judgments are based on the perception of informativeness concerning the query and its underlying user intent. Standard metrics for assessment, like precision, recall and nDCG (Järvelin and Kekäläinen, 2002), are not applicable to our setting. The notion of coverage pursued in this work refers to the yield of structured outputs by RE systems rather than document relevance. For example, a query-topic-wise highly relevant document that contains few extractable facts about named entities would still have low RE coverage.

Relevance of Coverage Estimates. Understanding and incorporating document coverage prediction into NLP-based information extraction is essential for several reasons. For *resource-bounded KB construction*, it is crucial to know which documents are most promising for extraction with limited budgets for crawling and RE processing and/or human annotation (Ipeirotis et al., 2007; Wang et al., 2019). For *claim refutation*, coverage estimates can help to assess statements as questionable if documents with high coverage do not support them. So far, claim evaluation systems mostly rely on textual cues about factuality or source credibility (Nakashole and Mitchell, 2014; Rashkin et al., 2017; Thorne et al., 2018; Chen et al., 2019).

For *question answering* over knowledge bases, it is important to know whether a KB can be relied upon in terms of complete answer sets (Darari et al., 2013; Hopkinson et al., 2018; Arnaout et al., 2021). Current coverage estimation techniques for KBs do this analysis only post-hoc after the KB is fully constructed (Galárraga et al., 2017; Luggen et al., 2019), losing access to valuable information from extraction time.

3.10 Summary

This chapter introduced the novel task of document coverage prediction for web documents. We outlined a range of approaches including feature-based, statistical as well as neural models, for tackling this task. Our proposed model, Herb, combines document representations from pretrained language models with heuristic features. Herb showcases performance improvements across relations. We also released a large-scale labeled dataset, DoCo, containing entity-relation-document-coverage tuples for experimental study of our task. Moreover, we demonstrated the utility of coverage estimates in two important downstream applications: knowledge base construction and claim refutation. While this chapter focused on improving recall from external documents, the next chapter shifts to a complementary perspective: extracting multi-valued relational knowledge directly from language models.

4

Information Extraction for Multi-Valued Relations from Language Models

4.1 Introduction

In the previous chapter, prioritizing highly informative documents increased extraction yield. As language modeling advanced, this chapter considers a different setting: directly probing language models for multi-valued slot filling to extract multiple objects for a given subject and multi-valued relation.

4.1.1 Motivation and Background

This chapter addresses information extraction (IE) with language models (LMs). Specifically, given a query with explicit mentions of a subject and a multi-valued relation of interest, the goal is to directly query a pretrained LM to extract the corresponding objects. The setup follows the slot-filling paradigm in IE (Surdeanu, 2013; Louvan and Magnini, 2020; Weld et al., 2022), with the key distinction that the source of information is a language model itself, without additional supporting evidence in the prompt. The input takes the form of a cloze-style prompt containing a `[MASK]` token that the model must complete.

Petroni et al. (2019) showcased the potential of relation-specific probes for extracting implicit knowledge from latent language representations through the `LAMA` framework. But the feasibility of reliably materializing factual knowledge directly from LMs remains an open challenge (Razniewski et al., 2021; AlKhamissi et al., 2022; Li et al., 2025a). Building upon the `LAMA` framework, where an LM predicts the object in the slot for given a cloze-style prompt such as "Dante was born in `[MASK]`", several methods (Jiang et al., 2020b; Shin et al., 2020; Zhong et al., 2021; Qin and Eisner, 2021) have developed increasingly effective prompting strategies.

Importantly, all these methods implicitly assume the existence of a single correct object per (subject, relation)-pair and are evaluated using precision at rank 1. In reality, however, many relations are associated with multiple correct object entities. Figure 4.1 illustrates this scenario: using an LM as a black-box and the input prompt "Italy and `[MASK]` share a border.", Phase 1 shows the log-likelihood of various "country" type named-entities at the `[MASK]` position, and among these the green-highlighted entities represent all the valid objects.

Our work focuses on directly probing LMs for the *multi-valued slot-filling task*. In this setting, the LM is queried in a zero-shot manner, without any task-specific in-context examples. Each prompt includes a subject and a multi-valued relation, and the LM is expected to generate a candidate list of objects. A key challenge is that the prior knowledge on the number of correct objects is unknown. The goal, therefore, is to maximize recall while maintaining precision by applying different selection mechanisms on the generated candidates. Phase 2 in Figure 4.1 illustrates two such mechanisms: top-k and prob-x. Setting the optimal values for the respective parameters (e.g., $k = 5$ and $x = 4\%$) leads to the best balance between precision and recall.

Extracting all correct answers reliably is difficult due to several factors: (i) the popularity bias of LMs, which favors incorrect popular (head) entities over correct long-tail ones; (ii) uncalibrated probabilities of candidate tokens, which misaligns likelihood with factual correctness; and (iii) single-token masking limitations in prompts, which hinder multi-token entity extraction and disambiguation. As of 2022, when this work was carried out, encoder-type pretrained LMs such as BERT (Devlin et al., 2019) were highly prevalent for the slot-filling task. Therefore, this chapter focuses on such models, while a detailed discussion of recent advances is provided in Section 4.7.

4.1.2 Research Questions

Overall, this work addresses the following research questions:

RQ1: How to generate candidate list of objects from an existing LM such that more than one correct object entity can be extracted? (Section 4.3.1)

RQ2: Since the LM's probabilities alone do not provide a clear indication of the objects' factual validity (Jiang et al., 2021b; Holtzman et al., 2021), what selection mechanism can help in extracting multiple correct objects? (Section 4.3.2)

RQ3: How robust are the relation-specific prompt formulations and selection mechanisms across different types of LMs? (Section 4.6)

4.1.3 Approach and Contributions

We formulate the multi-valued slot-filling task as a rank-then-select problem based on LM's confidence. In a two-step process, we first probe an LM using several existing prompt types to generate a large pool of candidate objects. We then propose several supervised and unsupervised selection mechanisms that assess the ranking of objects to sample and retain accurate objects at the end.

We introduce new relation-specific discrete prompts for LM probing and compare them against existing discrete and continuous prompting techniques. The generated object lists are evaluated by their ranking order, and our prompts result in higher-quality lists than the state-of-the-art automated methods. We outperform the best baseline[1], SoftPrompts (Qin and Eisner, 2021), by approximately 5 percentage points on the three most challenging relations {(`chemical compound`, `has-parts`, `.`), (`country`, `has-official-language`, `.`), (`musician`, `plays-instrument`, `.`)}, while being competitive

[1]as of 2022

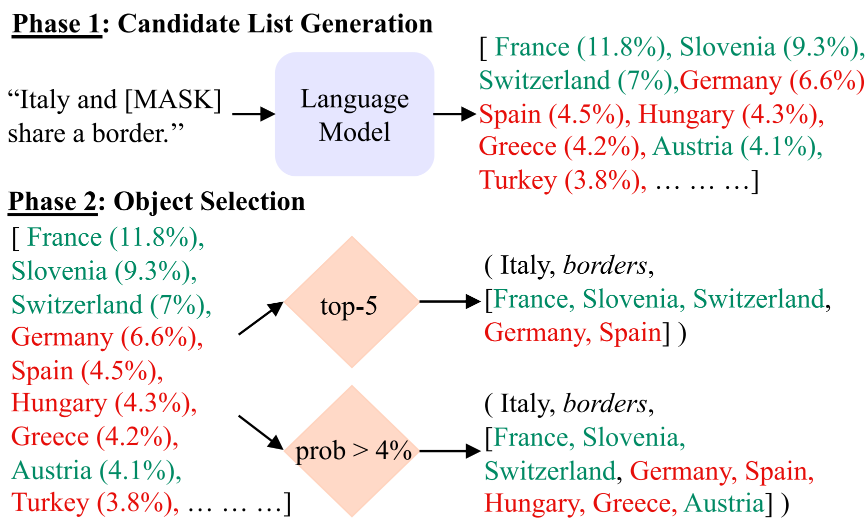

Figure 4.1: Probing LMs to extract objects for multi-valued relations.

on the other four relations {(`country`, `borders-country`, `.`), (`state`, `borders-state`, `.`), (`person`, `speaks-language`, `.`), and (`person`, `has-occupation`, `.`)}. While evaluating the outputs of the selection mechanisms, our best approach achieves 54.1% precision, 50.8% recall, and 49.5% F1-score. These empirical results demonstrate both the principal feasibility and the inherent difficulty of extracting *complete lists* of factual knowledge from internal LM representations, even today.

4.2 Problem Definition

Through language model probing, one can determine how much the model knows and if it can be used directly for a downstream task without additional fine-tuning. Chapter 2 provides a detailed background and formulation on LM prompting (Section 2.3.2) and probing (Section 2.3.3).

This work treats LMs as natural language generators, where an existing LM is probed using cloze-style prompts and the vocabulary tokens ranked by their corresponding LM probabilities are viewed as an open-ended candidate list of objects. Formally, given the subject s, relation r, and cloze-style prompt $f(r)$, the candidate list of objects O is $[o_1, o_2, \ldots,]$, such that $p(o_i) > p(o_{i+1})$, and $o_i \in \mathcal{V}$. Here $\mathcal{V}$ is the LM vocabulary and $1 \leq i \leq |\mathcal{V}|$. On this candidate list O, various selection mechanisms are proposed to get a subset of objects $\hat{O}$ such that the KB constructed via materialization of $\langle s, r, o \rangle$ knowledge triples is of highest quality.

4.3 Methodology

4.3.1 Candidate List Generation

When probing an LM using a cloze-style prompt, the model generates a probability distribution over vocabulary tokens to fill the masked position. For our task, we use prompts that mention a subject–relation pair $\langle$`s`, `r`$\rangle$ and consider the resulting ranked list of tokens, along with their corresponding probability scores, as candidate objects. In a zero-shot setting, the LM is probed using two categories of prompts: *discrete prompts* and *continuous prompts*.

Discrete Prompting. Methods for crafting discrete prompts (Li, 2023) use natural language tokens which are actual words or phrases, termed "hard" tokens, to form the prompt. These prompts are human-readable and interpretable, e.g., "Tisza river basins in `[MASK]`". Methods either search for or learn sequences of discrete tokens from the LM's vocabulary such that the prompt elicits best responses. Prompts are optimized via gradient-guided token search (Shin et al., 2020), paraphrase mining (Jiang et al., 2020b) or reinforcement learning (Deng et al., 2022).

Continuous prompting. Methods for continuous prompts (Chang et al., 2024a) learn vector representations or embeddings, termed "soft" tokens, to form the prompt. These embeddings are part of the model's input layer, do not correspond to real words, and are not easily interpretable by humans (Khashabi et al., 2022). They are continuously optimized through gradient-based approaches without altering the model's internal parameters (Zhong et al., 2021; Qin and Eisner, 2021).

Ours. We propose a collection of carefully designed prompts for generating the candidate list of objects for multi-valued relations. For each relation, we create 50 diverse relation-specific prompts by incorporating domain knowledge, relation types, and variations in sentence structure and grammar. Our prompts differ in verb form, tense, and the placement of the masked token (whether it appears in a prefix, suffix, or cloze-style position), as well as in the presence or absence of a period and object type in the context. The rationale behind these variations is inspired by prior work (Jiang et al., 2020b) on paraphrasing prompts, but specifically tailored for multi-valued information extraction with LMs.

4.3.2 Selection Mechanisms

Given a subject–relation pair $\langle \mathtt{s}, \mathtt{r} \rangle$, the probing step in Section 4.3.1 yields a ranked candidate list $O = \{(o_1, p_1), (o_2, p_2), \ldots, (o_n, p_n)\}$, where o_i is a candidate object and p_i, such that $p_1 \geq p_2 \geq \cdots \geq p_n$, is the model's probability for filling the masked position with o_i. For multi-token objects, we iteratively probe the model and aggregate token-level scores into an object-level score, e.g., compute the product of conditional token probabilities or sum of log-probs over the object tokens. The exposition below assumes such aggregation has been performed. We now describe parameterized mechanisms that lead to a validated subset of $(\mathtt{s}, \mathtt{r}, \mathtt{o})$ triples.

Running example. Let the subject be *Germany* and the relation be *shares-border*. A single cloze prompt *"Germany and `[MASK]` share a border."* produces the following candidate list O[2]:

Note that the ground-truth border set is {*Austria*, *Belgium*, *Czech Republic*, *Denmark*, *France*, *Luxembourg*, *Netherlands*, *Poland*, *Switzerland*}. The model's top ranks includes false positives such as *Italy*, *Russia* and so on (marked in red).

Top-k. The most probable k objects, termed top-k sampling in the context of text generation (Radford et al., 2019), are selected: $\hat{O} = \{o_1, \ldots, o_k\}$.
Example: Using the running example, with $k = 5$, output is $\{Austria, France, Italy, Poland, Switzerland\}$.
Strength: Simple and intuitive fixed-budget extraction, with k being the hyperparameter.

[2] using the BERT-base model https://huggingface.co/google-bert/bert-base-uncased?text=Germany+and+%5BMASK%5D+share+a+border.

Rank	Object	Probability	Cumulative Prob.
1	austria	0.168	0.168
2	france	0.136	0.304
3	italy	0.099	0.403
4	poland	0.094	0.497
5	switzerland	0.071	0.568
6	denmark	0.039	0.607
7	russia	0.035	0.642
8	sweden	0.031	0.673
9	norway	0.031	0.704
10	belgium	0.031	0.735
11	turkey	0.018	0.753
12	hungary	0.017	0.770

Limitation: This mechanism is sensitive to rank errors around the cutoff: can include high-rank false positives (e.g., *Italy*) and exclude low-rank true positives (e.g., *Denmark*, *Belgium*). Also, it is unsuited for relations with high variance, e.g., Iceland has zero neighbors, while Germany has nine.

Prob-x. Objects with a probability greater than or equal to x are chosen: $\hat{O} = \{\, o_i \; : \; p_i \geq x \,\}$.
Example: Using the running ex., with $x = 0.08$, output is $\{Austria, France, Italy, Poland\}$.
Strength: This mechanism adapts the output set size to model confidence, while filtering out the tail generations. Here, x is a hyperparameter.
Limitation: Low confidence true positives below x get discarded, which usually happens with long-tail object entities (e.g., *Switzerland*, *Denmark* and *Belgium*).

Cumul-x. Retain all objects, in order of probability, whose summed probability is no larger than x: $\hat{O} = \{o_1, \ldots, o_m\}$ where $\sum_{i=1}^{m} p_i \leq x$, and x is a hyperparameter. Unlike Prob-x, it would enable retaining candidates of similarly moderate probability.
Example: Using the running ex., with $x = 0.6$, the candidate object not exceeding 0.6 is at rank 6, so the output is $\{Austria, France, Italy, Poland, Switzerland, Denmark\}$.
Strength: This mechanism focuses on confidence-mass based budgeting. It is similar to nucleus top-p selection, where the key idea is to use the shape of the probability distribution to determine the set of tokens to be sampled from (Holtzman et al., 2020).
Limitation: The output size varies based on the distribution sharpness and can lead to exclusion of low-probability correct objects.

Count Probe. We probe the LM again for the cardinality n, number of objects, for ⟨`s`, `r`⟩, using an auxiliary count prompt such as: *"Germany borders `[MASK]` other countries."*. From the token distribution at `[MASK]`, we take the highest-probability integer token (either numerical or alphabetical, e.g., "`seven`") and parse it as n. We then subset the original candidate list by keeping the top n objects: $\hat{O} = \{o_1, \ldots, o_n\}$.
Example: The above count probe leads to the following candidate objects along with their likelihood:

{("several", 0.136), ("many", 0.130), ("with", 0.101), ("seven", 0.054), ("two", 0.044), ... }, using the BERT-base model[3]. Then $n = 7$, and the output is: {*Austria*, *France*, *Italy*, *Poland*, *Switzerland*, *Denmark*, *Russia*}.
Strength: This mechanism enforces a task-specific cardinality prior on the candidate list.
Limitation: If n is under-estimated, true objects in the tail are dropped, while if over-estimated, more false positives are admitted. Also, later works showed that LMs are poor at directly predicting count cardinalities (Ghosh et al., 2023), and uncalibrated likelihoods of encoder-type LMs makes the case even harder (Zhao et al., 2021).

Verification Probe: We probe the same LM again on each candidate object to factually verify the generated subject-relation-object $\langle \texttt{s}, \texttt{r}, \texttt{o} \rangle$ triple. A verification probe takes the form of a binary cloze question such as: *"Germany and Italy share a border? Answer: `[MASK]`"*. We compare the relative probabilities of the "`yes`" and "`no`" tokens in the masked position, by defining the following margin: $m(o_i) = p_{\texttt{yes}}(o_i) - p_{\texttt{no}}(o_i)$. We accept the candidate object o_i iff $m(o_i) > \alpha$, where α is a hyperparameter.
Example: Using $\alpha = 0.2$ on the candidate objects, the output is an empty list "[]".
Strength: This mechanism performs precision-oriented scrutinization by turning candidates into targeted factual checks. Prior work by Schick and Schütze (2021a) demonstrates the effectiveness of reformulating input prompts into pattern-verbalizer pairs and using likelihood of `yes`/`no` tokens for label classification. However, this was done for much simpler cases involving inputs with single correct answers and relied on semi-supervised model tuning.
Limitation: The calibration of `yes`/`no` token priors (e.g., via bias subtraction) has a direct effect on the final output and can lead to aggressive pruning due to uncalibrated likelihoods of LMs under the zero-shot probing setting.

4.4 Experimental Setup

4.4.1 Dataset

We select seven diverse multi-valued relations from the `LAMA` benchmark (Petroni et al., 2019). For each relation, we sample 200 subjects along with their complete list of objects from the Wikidata knowledge base (KB) (Vrandečić and Krötzsch, 2014). The subjects were picked based on popularity (head entities), measured using Wikidata ID and count of Twitter followers for person-type subjects. The dataset is publicly available at `https://github.com/snehasinghania/multi_valued_slot_filling`. Table 4.1 summarizes the statistics of the constructed dataset.

4.4.2 Evaluation Metrics

In the *ranking* phase, the quality of a candidate list is assessed using the maximally possible F1, termed max-F1, defined as the highest possible F1-score achieved by applying the top-k selection mechanism with the optimal k value. The optimal k is found by iterating over all possible choices

[3]`https://huggingface.co/google-bert/bert-base-uncased?text=Germany+borders+%5BMASK%5D+other+countries.`

Relation	Type	#Subject	Range (#Objects per Subject)
has-parts	compound $\rightarrow$ element	200	[2, 6]
borders-country	country $\rightarrow$ country	185	[1, 17]
official-language	country $\rightarrow$ language	196	[1, 16]
plays-instrument	person $\rightarrow$ instrument	200	[1, 27]
speaks-language	person $\rightarrow$ language	200	[2, 8]
has-occupation	person $\rightarrow$ occupation	200	[7, 20]
borders-state	state $\rightarrow$ state	200	[1, 14]

Table 4.1: Dataset Statistics

of k and calculating the respective F1-score on the subset of candidate objects and ground-truth objects. Formally, let k be the optimal threshold for the candidate list $\mathcal{O}$, generated for a subject s and relation r, and $\mathcal{GT}$ be the set of ground-truth objects for $\langle s, r \rangle$. Then max-F1 is:

$$\text{max-F1}(\mathcal{O}, s, r) = F1(\{o_1, \ldots o_k\}, \mathcal{GT})$$

Here, F1 is the harmonic mean of precision and recall. The max-F1 scores are macro-averaged across all the subjects for a chosen relation, and further used to compare the candidate lists generated by the prompting techniques.

In the *selection* phase, the output triples obtained after applying a selection mechanism are evaluated using precision, recall, and F1-score.

4.4.3 Baselines for Generation Phase

We use the following set of discrete and continuous prompting methods as baselines for generating the candidate list of objects.

LPAQA (Jiang et al., 2020b) used text mining, prompt paraphrases, and ensemble modeling to design the best discrete prompt for a given relation. They collected additional triples from Wikidata to tune the prompt template and hyper-parameters. The publicly released code and prompts[4], using both mining and paraphrased techniques, are used in our task.

AutoPrompt (Shin et al., 2020) method proposed a statistical model to automatically construct prompts by finding trigger words using gradient-based search algorithm (Wallace et al., 2019). They achieved better performance compared to the `LAMA` probe and LPAQA on the fact retrieval task and also used additional data from the TRex (ElSahar et al., 2018) dataset to tune the parameters. As mentioned in Shin et al. (2020), we used the five trigger tokens setup for an optimal prompt generation using the publicly released code and prompts[5].

OptiPrompt (Zhong et al., 2021) initializes prompt words as continuous vectors, which is further optimized in the embedding space using a training split from TRex (ElSahar et al., 2018) dataset.

[4] `https://github.com/jzbjyb/LPAQA`
[5] `https://github.com/ucinlp/autoprompt`

The vectors can be either randomly initialized, or manual prompts could be used as a starting point. We trained and optimized the OptiPromptusing author-released data and code[6] in two ways: one using their default initialization, and using our proposed best prompts.

SoftPrompts (Qin and Eisner, 2021) introduces soft-prompts consisting of "soft words" for convenient optimization and expressiveness by emphasizing particular or specific dimensions of words. In contrast to LPAQA, a mixture of weights model is used over the prompt templates obtained specific to a relation, forming a distribution over the soft-prompts. We trained and optimized the SoftPromptsusing author-released data and code[7] in two ways: one using their default initialization, and using our proposed best prompts.

4.4.4 Setup

We reuse the best prompts reported by each prompting baseline, which are tuned on much larger data. We probe BERT[8] (Devlin et al., 2019) on each ⟨s, r⟩ and generate the 500 most probable candidate objects. The generated list is post-processed to remove stopwords and other type-irrelevant objects depending on the relation type, only to retain sensible candidate objects. Our dataset is split into train, dev, and test, with 100/50/50 subjects per relation, for tuning and estimating parameters in the selection mechanisms.

4.5 Results

4.5.1 Candidate List Generation

We compare the candidate lists generated by each prompting method in Table 4.2. Our method generates the best object lists in terms of max-F1 score. In comparison to our prompts, discrete prompts (LPAQA variants and AutoPrompt) have a lower performance, while continuous counterparts (OptiPrompt and SoftPrompts) have a similar performance. Surprisingly, OptiPrompt obtained by initializing its continuous vectors using our prompts has a lower score.

To validate the effectiveness of our prompts and inspect if optimizing prompts on precision@1 is sufficient for extraction on multi-valued relations, we compare the best prompts in terms of precision@1 and max-F1 scores. In Tables 4.3 and 4.4, we observe that prompts performing well on precision@1 are not necessarily the best in terms of max-F1. Overall, prompts suitable for multi-valued extraction are more often of the prefix type, whereas prompts for single-object cases exhibit greater variance.

Furthermore, Table 4.5 presents examples of generated lists, where we observe that correct and incorrect objects are distributed unevenly across samples. In the Appendix (Section A.1) part of this thesis, Tables A.1 to A.7 lists all the prompt templates along with their corresponding scores. We notice inconsistencies in performance as prompts are subtly modified, highlighting the sensitivity of prompt formulation on LM probing. Moreover, the large gap between precision@1 and max-F1

[6] https://github.com/princeton-nlp/OptiPrompt
[7] https://github.com/hiaoxui/soft-prompts
[8] https://huggingface.co/google-bert/bert-large-uncased

⟨`subject, relation`⟩	LPAQA-M	LPAQA-P	AUTO	OPTI	SOFT	OURS
compound, *has-parts*	38.1	29.5	51.4	68.1	71.6	**78.5**
country, *borders-country*	66.4	64.9	71.6	73.2	**75.6**	72.8
country, *official-language*	81.9	75.2	71.8	75.9	79.9	**83.6**
person, *plays-instrument*	**63.4**	61.3	61.7	52.7	57.0	62.5
person, *speaks-language*	69.1	41.1	52.8	71.5	69.0	**72.8**
person, *has-occupation*	40.2	**44.9**	37.5	36.6	36.9	33.2
state, *borders-state*	23.8	24.3	24.4	25.9	**25.9**	25.7
Overall (avg.)	54.7	48.7	53.0	57.7	59.4	**61.3**

Table 4.2: max-F1 based comparison of candidate lists generated by probing the BERT model. LPAQA-M is LPAQA's mining-based prompts and LPAQA-P is LPAQA's paraphrasing-based prompts. AUTO is AUTOPROMPT, OPTI is OPTIPROMPT and SOFT is SOFTPROMPTS.

⟨`subject, relation`⟩	Our Prompts with best precision@1	hits@1
compound, *has-parts*	`[X]` contains `[MASK]` atom	78.50
country, *borders-country*	`[X]` and `[MASK]` share a border	84.86
country, *official-language*	People of `[X]` mostly speak in `[MASK]`.	93.37
person, *plays-instrument*	Musician `[X]` plays `[MASK]`.	67.50
person, *speaks-language*	In which language can `[X]` talk? Answer: `[MASK]`.	92.50
person, *has-occupation*	`[X]` is a well-known `[MASK]`.	59.00
state, *borders-state*	`[MASK]`, which is a `[Y]`, borders `[X]`.	37.50

Table 4.3: Our best performing prompts on precision@1 (%). The `[Y]` slot takes the object-type information, e.g., in ⟨state, *borders-state*⟩, `[Y]` could be "state", "governate" etc.

indicates the difficulty in designing task-specific prompts, and emphasizes the need for more robust and generalizable prompt construction strategies.

4.5.2 Object Selection

The candidate objects retained after applying a selection mechanism are compared against the ground-truth objects. The results are shown in Table 4.6. Among the evaluated methods, the top-k approach achieves the best overall F1-score, which is the macro-average of individual ⟨`s, r`⟩ tuple-specific F1-scores. However, the individual F1-scores and max-F1 (upper bound) have a large gap. This is because the probabilities of predicted tokens are not calibrated enough to match the actual factuality of the ⟨`s, r, o`⟩ triple.

Table 4.7 lists the best-performing prompt templates (based on max-F1) and their corresponding learned parameters for each selection mechanism. We also observe notable variation in the optimal prompts depending on the chosen mechanism. Such brittleness and inconsistency in prompt behavior have been reported in parallel studies as well (Elazar et al., 2021; Tam et al., 2023).

⟨`subject`, `relation`⟩	Our Prompts with best max-F1	max-F1
compound, *has-parts*	`[X]` has `[MASK]`, which is an atom.	78.52
country, *borders-country*	`[X]` and `[MASK]` share a border.	72.82
country, *official-language*	`[MASK]` is the main language of `[X]`.	83.57
person, *plays-instrument*	`[X]` plays `[MASK]`, which is an instrument	62.45
person, *speaks-language*	`[X]` speaks in `[MASK]`.	72.78
person, *has-occupation*	`[X]` is a well-known `[MASK]`	33.21
state, *borders-state*	`[X]` and `[MASK]` share a border	25.71

Table 4.4: Our best prompts among the 50 relation-specific prompts on max-F1 (%).

4.6 Analysis and Discussion

Although BERT was probed for 500 objects when generating object lists, only 119.7 objects were retained after post-processing. Objects with invalid types occur due to the zero-shot setting. Also, other eminent masked LMs, including BERT-base, RoBERTa-base, and RoBERTa-large (Liu et al., 2019), achieve 60.61%, 54.82%, and 58.90% max-F1 scores.

The max-F1 scores in Tables 4.2, 4.3 and 4.4 are far from 100%, i.e., LMs do not generate candidate lists that correctly rank all true objects above the false ones. In particular, max-F1 will not reach 100% when correct objects are ranked too low or absent. We found that 26.90% of valid objects in a candidate list were ranked below the optimal threshold and 27.75% of valid objects were not generated at all.

In Table 4.6, the top-k and prob-x achieve balanced precision and recall scores. The count-probe achieves a high recall since almost always a count greater than 10 is predicted, and in our dataset, the average count of ground-truth objects across all ⟨`s`, `r`⟩ is in [1,10] range. In the verification probe, the parameter α is near zero for most relations, and the probability of "`yes`" is greater than "`no`", leading to a selection of all the candidate objects. Although Schick and Schütze (2021a) and others show the effect of verbalizing labels to "`yes`" and "`no`" tokens in the few-shot setting on classification and inference tasks, optimally using them for multi-valued relation extraction remains an open challenge.

4.6.1 Effect of Prompt Template

In Table 4.6, each selection mechanism is evaluated on the generated candidate list using prompt templates shown in max-F1 column in Tables 4.3 and 4.4. However, by choosing a different set of prompts, a higher overall F1 score of 51.3% with a lower 59.9% max-F1 can be achieved by using the prob-x method, with Table 4.8 presenting the results. This change in F1 scores shows the hardness of designing robust prompts.

4.6.2 Effect of Relation Type

Candidate lists generated for popular subjects tend to achieve higher precision and recall. For instance, the F1 score for the relation (state, *borders-state*) with the top-k selection mechanism is the

⟨subject, relation⟩	Generated Object List	|GT|
(compound, *has-parts*)		
Calcium Carbonate	carbon (0.47), hydrogen (0.03), oxygen (0.02), calcium (0.01), silicon (0.01), nitrogen (0.002), sulfur (0.002)	3
Dopamine	hydrogen (0.09), nitrogen (0.05), carbon (0.05), oxygen (0.05), calcium (0.02), sodium (0.011), sulfur (0.009)	4
Sodium Chloride	carbon (0.15), hydrogen (0.09), oxygen (0.03), silicon (0.01), nitrogen (0.01), sulfur (0.006), sodium (0.006)	2
Thiocyanic Acid	hydrogen (0.1), carbon (0.1), oxygen (0.03), nitrogen (0.02), sulfur (0.01), silicon (0.003), sodium (0.002)	4
Water	oxygen (0.17), hydrogen (0.17), carbon (0.05), nitrogen (0.02), sodium (0.01), mercury (0.004), sulfur (0.003)	2
(country, *borders-country*)		
Germany	poland (0.14), austria (0.12), france (0.09), italy (0.06), belgium (0.05), russia (0.04), switzerland (0.04)	9
India	pakistan (0.34), bangladesh (0.19), myanmar (0.1), nepal (0.1), china (0.05), iran (0.02), bhutan (0.02)	8
Palau	japan (0.06), indonesia (0.05), taiwan (0.04), fiji (0.03), china (0.02), australia (0.02), philippines (0.01)	3
Malta	gibraltar (0.12), italy (0.11), cyprus (0.07), ireland (0.06), greece (0.05), tunisia (0.04), serbia (0.04)	1
Singapore	malaysia (0.7), thailand (0.1), indonesia (0.1), vietnam (0.02), myanmar (0.02), china (0.01), taiwan (0.01)	2
(country, *official-language*)		
Algeria	french (0.47), arabic (0.4), spanish (0.04), english (0.03), algerian (0.007), italian (0.005), latin (0.003)	2
Bolivia	spanish (0.9), english (0.07), portuguese (0.01), french (0.006), arabic (0.003), italian (0.002), latin (0.001)	4
Ethiopia	somali (0.52), arabic (0.08), english (0.05), ethiopian (0.04), italian (0.02), spanish (0.01), french (0.01)	1
Singapore	english (0.7), malay (0.18), chinese (0.03), tamil (0.03), mandarin (0.02), indonesian (0.005), arabic (0.003)	4
South Africa	english (0.9), dutch (0.03), french (0.03), portuguese (0.02), spanish (0.01), german (0.006), arabic (0.004)	11
(person, *plays-instrument*)		
A. R. Rahman	guitar (0.29), flute (0.18), piano (0.16), violin (0.08), saxophone (0.05), harmonica (0.04), clarinet (0.03)	16
Andy Hurley	guitar (0.36), piano (0.11), bass (0.06), violin (0.05), cello (0.04), accordion (0.03), drums (0.03)	1
Bruce Springsteen	guitar (0.54), piano (0.08), bass (0.05), drums (0.04), mandolin (0.03), harmonica (0.03), trumpet (0.03)	3
Owen Pallett	guitar (0.34), piano (0.15), violin (0.06), bass (0.05), cello (0.05), trumpet (0.03), drums (0.03)	4
Rino Sashihara	guitar (0.2815), flute (0.12), piano (0.12), violin (0.07), cello (0.04), accordion (0.04), clarinet (0.03)	1
(person, *speaks-language*)		
Alessandra Ambrosio	italian (0.9), english (0.1), spanish (0.02), french (0.02), german (0.01), portuguese (0.01), latin (0.01)	3
Amy Jackson	english (0.6), spanish (0.1), french (0.1), japanese (0.03), german (0.03), russian (0.02), italian (0.02)	4
Gustavo Petro	spanish (0.7), english (0.2), italian (0.03), portuguese (0.03), french (0.02), german (0.01), catalan (0.006)	4
Gad Elmaleh	english (0.4), arabic (0.4), french (0.13), hebrew (0.03), spanish (0.01), persian (0.007), russian (0.007)	4
Petro Poroshenko	russian (0.4), ukrainian (0.3), english (0.13), polish (0.05), belarusian (0.03), bulgarian (0.006), german (0.006)	6
(person, *has-occupation*)		
Donald Trump	politician (0.0005), american (0.0005), speaker (0.0004), name (0.0003), personality (0.0003), person (0.0003)	17
Neil Gaiman	author (0.001), writer (0.001), novelist (0.0003), artist (0.0003), contributor (0.0002), character (0.0002)	11
Richard Dawkins	author (0.0024), biologist (0.0019), writer (0.0018), psychologist (0.001), philosopher (0.001), scientist (0.0008)	15
George R. R. Martin	author (0.0032), historian (0.0025), writer (0.0013), scholar (0.0012), biologist (0.0005), novelist (0.0004)	10
Yoko Ono	artist (0.001), singer (0.001), musician (0.0004), actress (0.0003), author (0.0002), writer (0.0002), painter (0.0002)	10
(river, *basins*)		
Aras River	russia (0.18), uzbekistan (0.08), azerbaijan (0.07), armenia (0.06), kazakhstan (0.04), iran (0.04), ukraine (0.03)	4
Draa River	somalia (0.07), ethiopia (0.07), afghanistan (0.02), turkey (0.02), egypt (0.02), algeria (0.02), morocco (0.02)	1
Mekong River	vietnam (0.2001), cambodia (0.19), thailand (0.05), laos (0.03), china (0.01), myanmar (0.003), cameroon (0.003)	6
Limpopo River	botswana (0.25), zambia (0.16), namibia (0.13), zimbabwe (0.08), mozambique (0.07), angola (0.02), africa (0.02)	4
Jordan River	jordan (0.33), israel (0.06), syria (0.05), iraq (0.02), iran (0.02), egypt (0.02), palestine (0.02), lebanon (0.01)	6
(state, *borders-state*)		
Alabama	mississippi (0.4), georgia (0.3), tennessee (0.1), louisiana (0.05), florida (0.04), arkansas (0.02), texas (0.02)	4
Castile and León	navarre (0.26), galicia (0.22), catalonia (0.14), aragon (0.04), castile (0.02), valencia (0.01), mexico (0.003)	10
La Rioja Province	mendoza (0.05), navarre (0.05), galicia (0.02), madrid (0.01), catalonia (0.007), piedmont (0.006)	5
Gelderland	utrecht (0.40), holland (0.06), hesse (0.02), hamburg (0.01), jersey (0.003), bremen (0.002), berlin (0.001)	7
Fukushima Prefecture	tokyo (0.23), hiroshima (0.14), okinawa (0.13), kyoto (0.1), osaka (0.03), nagoya (0.03), saga (0.01)	6

Table 4.5: Samples of generated object list for five unique subjects on multi-valued relations. The green highlighted valid objects, while the red ones are wrong. The |GT| column gives the total no. of ground-truth objects for the corresponding subject.

⟨subject, relation⟩	top-k			prob-x			cumul-x			count-probe			verify-probe			max-F1
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
compound, *has-parts*	62.5	78.1	68.0	60.3	76.4	65.4	37.8	74.7	37.8	54.2	78.7	61.8	16.1	70.7	22.9	78.5
country, *borders-country*	64.0	55.4	54.2	63.4	58.7	55.4	56.0	58.4	46.5	21.9	71.7	30.5	1.9	68.2	3.6	72.8
country, *official-language*	96.0	74.1	80.1	94.0	75.8	80.5	52.8	71.3	43.2	28.9	82.3	40.5	27.0	32.4	4.9	83.6
person, *plays-instrument*	46.0	42.3	38.8	51.7	40.8	39.1	51.8	41.4	33.8	18.2	60.9	25.5	7.1	24.4	4.8	62.5
person, *speaks-language*	52.5	59.8	55.1	69.6	56.7	60.0	56.2	57.3	46.8	37.4	69.0	47.3	3.5	53.0	6.4	72.8
person, *has-occupation*	33.3	23.5	27.3	3.2	85.1	6.1	30.1	36.2	18.6	23.1	30.0	25.9	5.5	41.7	9.0	33.2
state, *borders-state*	24.4	22.6	22.9	63.1	21.1	24.9	21.9	18.3	13.7	10.0	24.3	13.9	2.4	26.2	4.3	25.7
Overall (averaged)	54.1	50.8	**49.5**	**57.9**	59.2	47.4	43.8	51.1	34.4	27.7	**59.5**	35.1	9.1	45.3	8.0	61.3

Table 4.6: Results on comparing triples using precision, recall, and F1-score when probing the BERT model and applying a selection mechanism. The bold-faced numbers are the highest achieved precision, recall, and F1 scores.

lowest, largely due to the presence of long-tail subjects. Also, in relations with a large possible set of valid objects, such as *occupation*, the LM only generates common professions with a high probability. This behavior negatively affects the F1-score by overlooking less frequent but correct objects.

Interestingly, the bias towards towards common entities benefits language-type relations, such as (*person, speaks-language*) and (*country, official-language*). In these cases, higher recall is achieved because the ground-truth datasets for such relations are themselves biased toward widely spoken languages like English, French, and Spanish. A similar variation in performance across relation types has also been reported in prior works (Shin et al., 2020; Zhong et al., 2021).

4.6.3 Calibration using Web Signals

Prior work uses search-engine *hit counts* as a signal to check factuality or to pick the best answer in query-reformulation based question-answering systems (Cilibrasi and Vitányi, 2004; Vitányi and Cilibrasi, 2010; Kwok et al., 2001). Following the same rationale, we use Bing search engine's estimated hit count (the number of matching results) to calibrate and select objects from the candidate list. Concretely, Bing receives each ⟨s, r, o⟩ as a natural-language query. Using the running example from Section 4.3.2, the Bing query for the candidate object "austria" is *"Germany borders Austria"*.

For the candidate list $O = \{o_1, o_2, \ldots, o_n\}$, let $h(o_i)$ denote the Bing hit count returned for candidate object o_i. Also, let $r(o_i) \in \{1, 2, \ldots\}$ be the baseline rank of o_i in the candidate list (where rank 1 is the best). We use the web signal in two ways:

1. ***subset*** (filtering): keep only objects with $h(o_i) > 0$, i.e., objects with non-zero web evidence.

2. ***rerank*** (calibrated reordering): for objects with non-zero web evidence $h(o_i) > 0$, compute a calibration score

$$s(o_i) = \frac{h(o_i)}{r(o_i)}$$

to weigh the candidates using the baseline ranking and count of web evidence. The candidates are reranked by $s(o_i)$ and these scores are converted to calibrated probabilities by normalizing them across the candidate list.

⟨subject, relation⟩	Metric	avg-cutoff	Our Prompts
compound, *has-parts*	top-k	4	[X] has [MASK], which is an atom
	prob-x	0.02	[X] has [MASK], which is an atom
	cumul-x	0.53	[X] has [MASK], which is an atom
	count-alpha	2.26	[X] consists of [MASK] elements.
	count-num	4.64	[X] consists of [MASK] elements
	verify-probe	$\alpha = 0.06$	[X] consists of [Y] atom. Is this correct? Answer: [MASK].
country, *borders-country*	top-k	3	[X] and [MASK] share a border.
	prob-x	0.05	[X] and [MASK] share a border.
	cumul-x	0.79	[X] and [MASK] share a border.
	count-alpha	2.54	[X] shares border with [MASK] countries.
	count-num	13.36	[X] shares border with [MASK] countries
	verify-probe	$\alpha = 0$	[X] and [Y] share a border. Is this correct? Answer: [MASK].
country, *official-language*	top-k	1	[MASK] is the main language of [X].
	prob-x	0.22	[MASK] is the main language of [X].
	cumul-x	0.91	[MASK] is the main language of [X].
	count-alpha	3.42	[X] has [MASK] official languages.
	count-num	2.18	[X] has [MASK] official languages
	verify-probe	$\alpha = 0.11$	[Y] is the official language of [X]. Is this correct? Answer: [MASK].
person, *plays-instrument*	top-k	2	[X] plays [MASK], which is an instrument
	prob-x	0.12	[X] plays [MASK], which is an instrument
	cumul-x	0.54	[X] plays [MASK], which is an instrument
	count-alpha	2.98	[X] plays [MASK] instruments.
	count-num	6.88	[X] plays [MASK] instruments
	verify-probe	$\alpha = 0.28$	[X] plays [Y]. Is this correct? Answer: [MASK].
person, *speaks-language*	top-k	4	[X] speaks in [MASK].
	prob-x	0.05	[X] speaks in [MASK].
	cumul-x	0.87	[X] speaks in [MASK].
	count-alpha	4.16	[X] speaks in [MASK] languages.
	count-num	6	[X] speaks in [MASK] languages.
	verify-probe	$\alpha = 0.24$	[X] can speak in [Y]. Is this correct? Answer: [MASK].
person, *has-occupation*	top-k	8	[X] is a well-known [MASK]
	prob-x	0	[X] is a well-known [MASK]
	cumul-x	0.01	[X] is a well-known [MASK]
	count-alpha	4.74	[X] had a total of [MASK] different professions.
	count-num	13.64	[X] had a total of [MASK] different professions.
	verify-probe	$\alpha = 0$	[X] is a well-known [Y]. Is this correct? Answer: [MASK].
state, *borders-state*	top-k	5	[X] and [MASK] share a border
	prob-x	0.04	[X] and [MASK] share a border
	cumul-x	0.75	[X] and [MASK] share a border
	count-alpha	3.16	[X] shares border with [MASK] states
	count-num	13.04	[X] shares border with a total of [MASK] states.
	verify-probe	$\alpha = 0$	[X] and [Y] share a border. Is this correct? Answer: [MASK].

Table 4.7: The prompt templates used for generating the object list. The avg-cutoff shows the learned parameters of each selection mechanism averaged across all subjects.

⟨subject, relation⟩	Our Prompts	avg-cutoff	Precision	Recall	F1 score	max-F1
compound, *has-parts*	[X] has [MASK], which is an atom.	0.02	60.33	76.40	65.41	78.52
country, *borders-country*	[X] has borders with [MASK].	0.07	74.73	55.41	58.45	71.41
country, *official-language*	The official language of [X] is [MASK].	0.15	94.33	79.97	83.38	83.54
person, *plays-instrument*	[X] likes to play the [MASK].	0.12	51.33	46.52	41.48	58.19
person, *speaks-language*	[X] speaks in [MASK].	0.05	69.60	56.72	59.99	72.78
person, *has-occupation*	[X] is a [MASK].	0.01	22.79	31.26	25.42	29.30
state, *borders-state*	[X] and [MASK] share a border	0.04	63.10	21.12	24.91	25.71
Overall			62.32	52.49	51.29	59.9

Table 4.8: Higher F1 score achieved by using a different set of our proposed prompts and prob-x mechanism.

⟨subject, relation⟩	Our Prompts	Prob-x			subset			rerank		
		P	R	F1	P	R	F1	P	R	F1
compound, *has-parts*	[X] has [MASK], which is an atom.	60.3	76.4	**65.4**	84.2	17.7	17.0	57.0	80.9	64.1
country, *borders-country*	[X] has borders with [MASK].	74.7	55.4	58.5	82.3	53.5	56.0	58.9	69.4	**58.8**
country, *official-language*	The official language of [X] is [MASK].	94.3	80.0	**83.4**	91.7	68.3	71.1	69.3	81.6	69.3
person, *plays-instrument*	[X] likes to play the [MASK].	51.3	46.5	**41.5**	83.0	30.1	33.8	27.7	53.2	32.4
person, *speaks-language*	[X] speaks in [MASK].	69.6	56.7	**60.0**	83.0	19.7	22.1	48.5	66.3	51.2
person, *has-occupation*	[X] is a [MASK].	23.0	31.3	**25.4**	19.7	23.7	20.3	19.8	24.1	20.5
state, *borders-state*	[X] and [MASK] share a border	63.1	21.1	**24.9**	93.0	18.3	22.3	69.0	21.9	23.4
Overall		57.9	59.2	**47.4**	76.7	33.1	34.7	50.1	56.8	45.7

Table 4.9: Results on calibrating candidate object probabilities with Bing hit rates

Compared to prob-x selection mechanism, the *subset* method increases precision and decreases recall, with a lower overall F1 score of 34.7%. On the other hand, the *rerank* method trades toward higher recall and lower precision with a similar overall F1 of 45.7%. All scores are reported in Table 4.9.

4.6.4 Effect of Language Model Size

We probed models larger than BERT-large (334M parameters) such as T5[9] (Raffel et al., 2020) (700M parameters) and BART[10] (Lewis et al., 2020a) (400M parameters), which can generate a list of tokens with likelihoods using Beam search decoding algorithm (Sutskever et al., 2014). With the top-k selection mechanism, T5 achieves 43.6% precision, 41.7% recall, and 40.3% F1 score. BART achieves an even lower 32.0% precision, 34.3% recall, and 30.8% F1 score. Table 4.10 gives all the scores.

These models tend to generate common objects and exhibit repetitive behavior. Also, using autoregressive models like GPT-3 (Brown et al., 2020), other works (Alivanistos et al., 2022; Cohen et al., 2023) extract multi-valued relations using in-context learning. However, unlike our method, with no control over the selection mechanism, the LM directly outputs one final list. While internally autoregressive models also use token probabilities that could be used for our approach, once a full list is generated, previously generated list items conflate the probabilities of items.

[9] https://huggingface.co/google-t5/t5-large
[10] https://huggingface.co/facebook/bart-large

⟨subject, relation⟩	Our Prompts	T5-large			BART-large		
		P	R	F1	P	R	F1
compound, *has-parts*	[X] has [MASK], which is an atom.	67.3	64.9	61.5	58.7	56.2	56.2
country, *borders-country*	[X] and [MASK] share a border.	43.8	53.1	45.0	44.8	60.4	47.3
country, *official-language*	[MASK] is the main language of [X].	82.0	61.1	66.8	0	0	0
person, *plays-instrument*	[X] plays [MASK], which is an instrument	13.3	14.8	12.6	40.0	36.1	33.5
person, *speaks-language*	[X] speaks in [MASK].	47.0	40.4	43.0	22.1	28.7	24.5
person, *has-occupation*	[X] is a well-known [MASK]	27.2	36.4	30.8	36.5	37.1	34.4
state, *borders-state*	[X] and [MASK] share a border	24.6	21.5	22.3	21.5	21.4	19.8
Overall		43.6	41.7	40.3	32.0	34.3	30.8

Table 4.10: Results on probing T5 and BART model with top-k selection mechanism

4.7 Language Models for Knowledge Base Construction

To explore the feasibility of constructing knowledge bases directly from LMs, the proposed task of probing LMs for multi-valued relation extraction, along with an extended version of the proposed dataset, was hosted as the *LM-KBC Challenge*[11] at the 21st International Semantic Web Conference (ISWC 2022). The challenge invited participants to *materialize* knowledge bases from LMs for a given set of subjects and relations drawn from our benchmark[12]. Unlike earlier probing benchmarks like LAMA (Petroni et al., 2019), LM-KBC adopted a more realistic setting of the multi-valued slot-filling task: given a subject and relation, systems had to return the complete list of object entities (possibly zero, one, or many). The challenge offered two tracks: (i) a restricted track, using only BERT as the choice of LM; (ii) an open track, allowing any LM.

The challenge gained a lot of traction from the semantic web community. Submissions across both tracks confirmed the potential, and the difficulty, of direct LM-based extraction. In the BERT track, all systems used variants of our probing approach, with extensive prompt engineering and candidate filtering. Notably, the system by Li et al. (2022a) additionally fine-tuned BERT under a masked language modeling (MLM) objective with the subject-relation-object triples and achieved an F1 score of 55%. In the open track, the system by Alivanistos et al. (2022), queried GPT-3 (Brown et al., 2020) via instruction-based, style-formatted few-shot prompting[13] with carefully crafted exemplars, and even issued follow-up verification prompts (yes/no questions) to check the generated facts, finally reaching 67.6% F1 score[14].

The challenge lead to several insights, aligning with the research questions in this chapter (Section 4.1):

1. *Prompt design is critical:* Nearly all teams invested in manual prompt engineering and even a small change in the prompt led to performance differences. This highlights the sensitivity of LMs to prompt formulation observed in our experiments as well.

[11] https://lm-kbc.github.io/challenge2022/
[12] https://github.com/lm-kbc/dataset2022
[13] https://github.com/HEmile/iswc-challenge
[14] https://codalab.lisn.upsaclay.fr/competitions/5815#results

2. *Relation-specific adaptation helps:* Fine-tuning the LM, or tailoring prompts/thresholds per relation outperformed one-size-fits-all settings. This suggests that the optimal selection mechanism depends on relation and how facts are stored in the LM's parametric knowledge.

3. *No-object cases remain challenging:* For about 10% of subjects that truly had no object, even the best systems hallucinated answers, underscoring the difficulty of answer abstention.

Overall, the LM-KBC challenge re-iterated the feasibility of zero-shot LM-based KB construction. It also exposed the above limitations and open challenges around prompt sensitivity, calibration, and the reliable handling of unanswerable queries. Subsequent editions (Razniewski et al., 2023, 2024) reported further gains in F1 score, up to almost 90%, by advances in modeling and generation capabilities of LMs—specifically on easy relations with popular (head) entities. The latest edition of the challenge (Razniewski et al., 2025) focuses heavily on no-object cases and numerical type objects, with results indicating that current systems are still far from perfect in the zero-shot setting.

4.7.1 Current Annecdotes

To gauge how close multi-valued extraction is to being "solved", in terms of both precision and recall, by current LLMs (e.g., GPT-5[15] and Gemini-2.5-Pro (Comanici et al., 2025)), we run a focused, illustrative probe using three representative examples. Since these LLMs perform external search, tool calling and reasoning for answer generation by default, we operate in the zero-shot setting[16].

Example 1: For the ⟨`Giza Governorate`, `state-borders`⟩ instance, we prompt the LLM with the query: *"Give a concise and complete list of states bordering the Giza Governorate."*
Ground truth: eleven governorates, including [*Beheira Governorate, Beni Suef Governorate, Cairo Governorate, Faiyum Governorate, Monufia Governorate, Qalyubia Governorate, Matrouh Governorate, Minya Governorate, New Valley Governorate, Red Sea Governorate, Suez Governorate*].
Outputs:

GPT-5: Here are the governorates bordering Giza Governorate in Egypt:

1. Beheira Governorate (north)
2. Monufia Governorate (north)
3. Qalyubia Governorate (north)
4. Cairo Governorate (east)
5. Suez Governorate (east)
6. Faiyum Governorate (south)
7. Beni Suef Governorate (south)
8. Minya Governorate (south)
9. New Valley Governorate (west & south-west)
10. Matrouh Governorate (west)

If you meant countries or a different kind of border (e.g., desert vs Nile valley), I can check and list those too.

[15] `https://openai.com/index/introducing-gpt-5/`
[16] Evaluation conducted on 28-10-2025

Gemini-2.5 Pro: Giza Governorate borders the following seven governorates:

1.Beheira (to the north)
2.Monufia (to the north)
3.Qalyubia (to the northeast)
4.Cairo (to the east)
5.Faiyum (to the south)
6.Beni Suef (to the south/southeast)
7.Matruh (to the west)
8.New Valley (El Wadi el Gedid) (to the south/southwest)

Giza is a large governorate, and its borders extend from the densely populated areas opposite Cairo, deep into the Western Desert, which is why it borders so many other regions.

Analysis: Both models achieve 100% precision but fall short on recall with GPT-5 at 90% and Gemini-2.5-Pro at 72.7%. Moreover, Gemini states *seven governorates* but lists eight, illustrating that text fluency can hide inconsistencies and be misleading. Despite this, these results are substantially stronger than BERT on the same $\langle \texttt{s}, \texttt{r} \rangle$ instance.

Example 2: For the ⟨`Brian Jones`, `plays-instrument`⟩ instance, we prompt the LLM with the query: *"Give a concise and complete list of instruments played by Brian Jones."*
Ground truth: 27 instrumments, including [*accordion, acoustic guitar, appalachian dulcimer, banjo, bass guitar, cello, clarinet, flute, guitar, slide guitar, harmonica, harp, horn, keyboard, mandolin, marimba, mellotron, oboe, piano, recorder, saxophone, sitar, tambura, theremin, trombone, trumpet, xylophone*]
Outputs:

GPT-5: Here is a concise list of instruments played by Brian Jones (founder of the The Rolling Stones):

1.Guitar (electric, acoustic, slide)
2.Harmonica
3.Keyboards (piano, organ)
4.Sitar
5.Marimba
6.Recorder
7.Saxophone
8.Mellotron
9.Dulcimer (Appalachian)
10.Autoharp
11.Oboe
12.Flute
13.Kazoo

If you like, I can provide a more exhaustive list (over 20 instruments) with references.

Gemini-2.5 Pro: Brian Jones was a notable multi-instrumentalist. Here is a comprehensive list of the instruments he played:

1.Guitars: Electric, acoustic, and slide guitar
2.Keyboards: Piano, organ, harpsichord, Mellotron
3.Wind: Harmonica, saxophone (alto and soprano), clarinet, flute, recorder, oboe
4.Strings: Sitar, Appalachian dulcimer, autoharp, cello
5.Percussion: Marimbas, glockenspiel, xylophone, castanets, tambourine, and other assorted percussion
6.Other: Trumpet, trombone, accordion
7.Vocals: Backing vocals

This video includes a discussion of Jones's musicianship, mentioning the marimbas and other instruments he brought to the band's sound.

Analysis: Both models attain high precision yet fall short on long-tail recall. Although the generations contain less-common instruments, other long-tail entities, such as *mandolin*, *tambura*, and *banjo*, are absent. Moreover, to achieve 100% recall, iterative prompting (follow-up prompt) is required, which weakens the zero-shot premise and complicates fair comparison.

Example 3: For the ⟨`Afghanistan`, `official-language`⟩, we prompt LLM with the query: *"Give a concise and complete list of official languages of Afghanistan."*
Ground truth: eight languages, including [*baluchi, dari, nuristani, pamir, pashayi, pashto, turkmen, uzbek*]
Outputs:

GPT-5: Here is the list of official languages of Afghanistan according to its 2004 Constitution (Article 16):

1.Pashto
2.Dari (Afghan Persian)

In addition, the Constitution states that in any region where a "majority of the people speak" one of the languages — Uzbeki, Turkmeni, Baluchi, Pashai, Nuristani or Pamiri — that language may also be used as a third official language in that region.

Gemini-2.5 Pro: The official languages of Afghanistan are:

1.Pashto
2.Dari (also known as Afghan Persian)

Note: Dari and Pashto are the two main national official languages. Historically, the constitution also recognized other languages like Uzbek, Turkmen, Balochi, Pashayi, and Nuristani as "third official languages" in areas where they are spoken by a majority.

Analysis: Both models return high-precision head items but fall short on recall. The responses are temporally anchored and implicitly interpret the scope of the prompt to be the current timeline (ignoring facts changed over time). Also, surface form and granularity issues (e.g., *Uzbeki* vs. *Uzbek*, *Baluchi*/*Balochi*, *Pamiri* as a language group) further complicate exact matching based evaluation.

Collectively, these insights show that even today's top-end LLMs cannot completely solve this task, and motivate the methodological developments in the next chapters (5 & 6).

4.8 Related work

While a detailed background on the relevant topics is given in Chapter 2, this section highlights prior work at the time of this project[17].

Eliciting factual knowledge directly from parametric memory of LMs was popularized by Radford et al. (2019); Petroni et al. (2019); AlKhamissi et al. (2022). Several follow-up approaches (Bouraoui et al., 2020; Goswami et al., 2020; Chen et al., 2022b) used relational metadata and textual corpora to fine-tune the LM using knowledge-enriched representations and efficiently extract knowledge. Jiang et al. (2020a); Kassner et al. (2021) focused on multilingual knowledge extraction, while Dhingra et al. (2022) looked at extracting temporal knowledge from LMs. However, almost all these approaches assume each query has a single correct answer and evaluate on precision@1. In reality, many relations are multi-valued, with several valid objects per subject. This gap motivates focus on methods that can effectively handle and verify multiple candidate answers for a given query.

A complementary line of work investigates how to design prompts that better elicit factual knowledge from LMs. Prompting strategies proposed by Jiang et al. (2020b); Shin et al. (2020); Zhong et al. (2021); Qin and Eisner (2021); Liu et al. (2023a) optimize prompts—either discretely or continuously—for improved knowledge extraction in cloze-style settings. Beyond manual or automatically discovered prompts, prefix-tuning (Li and Liang, 2021), P-Adapters (Newman et al., 2022), and related approaches (Shen et al., 2022) provide a lightweight alternative to full LM fine-tuning: the LM's weights are frozen and a small set of task-specific prefix vectors is learned to generation. With such learned prefixes, LM outputs can be steered for tasks like slot filling.

4.9 Summary

This chapter focused on probing language models directly for information extraction. We reviewed existing prompt engineering methods and proposed new relation-specific prompts for multi-valued relations. Our selection mechanisms enable more effective filtering of valid triples. The detailed performance analysis highlights both the strengths and limitations of using zero-shot probing for the multi-valued slot-filling task. We also provided insights into the behavior of current large language models. While these models are increasingly capable of recalling a larger set of correct objects, their performance remains limited—particularly for long-tail entities, temporally evolving facts, and no-object cases. Moreover, the lack of explainability persists and is only partially mitigated by augmenting the prompt with textual evidence, motivating the work presented in the next chapter.

[17] in the year 2023

5

Information Extraction from Long Documents

5.1 Introduction

In the previous chapter, extraction directly from language models did not achieve both high accuracy and high recall. In this chapter, we return to information extraction (IE) from text, similar in spirit to Chapter 3, but with a different focus. Instead of extracting from multiple corroborating documents, we address extraction from low-redundancy yet very long documents, again with the goal of maximizing recall.

5.1.1 Motivation and Background

IE is the task of distilling structured information from unstructured text. Specifically, relation extraction (RE) aims to yield subject-predicate-object (SPO) triples, where S and O are named entities that stand in a certain relation P. The state-of-the-art neural models perform well in terms of precision but have limited recall (Han et al., 2020; Zhao et al., 2024b; Qin et al., 2024). More recently, large language models (LLMs) perform well on RE benchmarks but exhibit deficits in recall, especially on long-tail facts (Kandpal et al., 2023; Veseli et al., 2023a; Sun et al., 2024). Moreover, most methods are designed to operate on single passages, as classifiers or sequence taggers, even when framed as a document-level RE task (Quirk and Poon, 2017; Yao et al., 2019). However, long documents—such as novels, stories, and literary nonfiction, or treating the web pages about a single target entity as one long document—pose unique challenges for IE.

Our work focuses on the underexplored setting with two *"longs"*: extracting a **long list** of object entities that stand in a certain relation to a subject, appearing in **long text**, such as entire books or websites with many pages. Examples include extracting a complete list of (nearly) all acquisitions and subsidiaries of Alphabet Inc., identifying all artists who have covered Bob Dylan songs, or finding all friends of Harry Potter in the Harry Potter book series.

To illustrate the problem, consider enemies/opponents of *Michael Corleone* in The Godfather books by Mario Puzo. Figure 5.1 shows book excerpts with cues about *McCluskey, Sollozzo, Roth, Tommasino and Fabrizzio* being in this list (which, according to sources like fan wikis, has 40 people). We observe three cases: easy (left), hard (middle), and challenging (right). The easy cases are salient entities that are frequently mentioned—extracting them needs only one or two informative passages, so that picking one or two passages about them may already be viable for proper extraction. The hard cases arise for entities that appear infrequently (like *Hyman Roth*, who is a minor figure

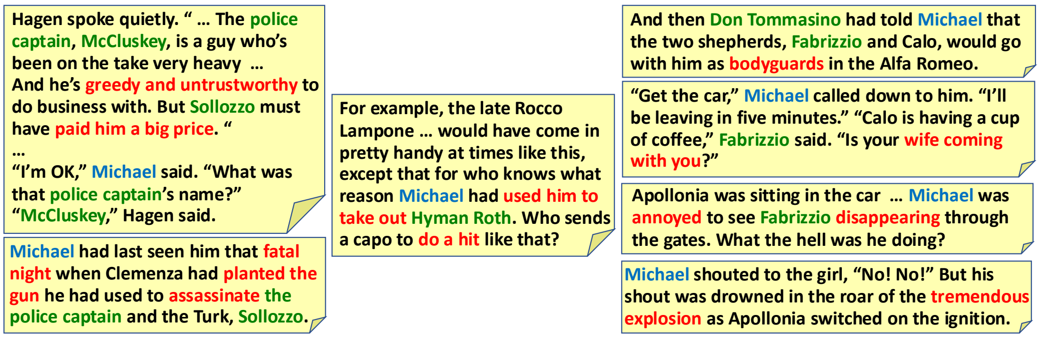

Figure 5.1: Example for the problem of long lists from long narratives. For the subject "Michael Corleone", we aim to extract all 40 enemies/opponents, appearing in the books.

in the books); here, the issue is finding the "needle-in-the-haystack" (Kamradt, 2023; Bai et al., 2024), so that finding the right passages is the main issue. Finally, the most challenging cases involve vague and terse cues for the predicate, requiring deeper inference over multiple, possibly scattered, passages—such as identifying *Fabrizzio* as the culprit behind the car bomb attack on Michael Corleone's wife.

Beyond factual knowledge bases, long-form IE is central to literary analysis. Understanding and analyzing narrative texts often involves entity markup and the extraction of relations between characters (Piper et al., 2021; Bamman et al., 2024). For instance, to discover narrative patterns in contemporary or historical fantasy stories, an RE system should track character movements across locations and label them by roles or sentiments (Wilkens et al., 2024). Similarly, cultural studies on gender roles in fiction across different epochs and regions (Silva et al., 2023; Kejriwal and Nagaraj, 2024) require labeling of character types and relationships. To support such analyses, tools for named-entity recognition, named-entity disambiguation and relational IE (a.k.a RE) must be adapted to the specifics of literary language and narrative structure.

There is ample work on RE, based on deep neural networks (Han et al., 2020; Zhao et al., 2024b). Recent methods employ LLMs for encoding input texts (Josifoski et al., 2022; Ma et al., 2023b; Xu et al., 2024a). These systems typically behave like sequence-to-sequence taggers: given text T and target subject S, they identify candidate objects O appearing in T, tag cue words for relation P, and classify each SPO candidate as valid or invalid. The key limitation is that texts are short—often single paragraphs, commonly from Wikipedia. Thus, there are only a few O candidates, and the task reduces to classification: mapping SO candidates onto none, one, or more predicate labels P.

More importantly, RE methods perform well when the S and O entities are salient, the text T is short, and the language style can be learned upfront via training on Wikipedia or fine-tuning on a specific corpus. However, when the input spans an entire book, pre-training has limited value and fine-tuning is infeasible due to the lack of annotated data. Also, the desired outputs would include long-tailed O's that appear only a few times over hundreds of pages. In contrast to the $SO \rightarrow P$ approach of standard RE, we cast this underexplored task as $SP \rightarrow \{O\}$: given subject S and relation P, extract/generate a long—ideally complete—list of objects O that stand in relation

P with S. Here, S and O are named entities and P is a relation such as `parent`, `family`, `friend`, `opponent` and so on.

5.1.2 Research Questions

This goal entails answering the following research questions:

RQ1: How well do LLMs perform on this challenge, and how much value is added by running LLMs in RAG mode? (Section 5.5.1)

RQ2: How can the outcome of LLM with RAG methods be further enhanced? (Section 5.5.2)

RQ3: How can we boost recall without losing too much in precision? (Section 5.5.3)

5.1.3 Approach

We devise a novel methodology to address this challenging task. Our method, called **L3X** (**L**M-based **L**ong **L**ist e**X**traction), operates in two stages:

Stage 1: Recall-oriented Generation. An LLM is prompted with the subject S and relation P at hand, and tasked to generate a full list of objects through various prompt formulations. In addition, we use information retrieval (IR) methods to find promising candidate passages from long texts and feed them into the LLM prompts. In contrast to prior works on retrieval-augmented LLMs, we retrieve a large number of such passages (e.g., 500 for a given SP pair) and judiciously select the best ones for prompting. Moreover, our method iteratively re-ranks the passages and re-prompts the LLM, to improve recall of initial generation of objects.

Stage 2: Precision-oriented Scrutinization. Given a high-recall list of object candidates from Stage 1, Stage 2 uses conservative techniques to corroborate or prune objects. We employ novel techniques to identify high-confidence objects and their best support passages, leverage cross-passage similarity to reassess lower-confidence candidates, and produce a final, precision-controlled list.

Since we tackle an unexplored task, we curated two datasets, covering fiction books and web documents, respectively. The books dataset, which is the primary target, consists of 11 books or book series, with a total of 16,000 pages. It covers 8 relations of long-tailed nature, including {`parent`, `child`, `sibling`, `family`, `friend`, `opponent`, `placeHasPerson`, and `hasMember`}. The second (web) dataset comprises approximately 10 million web documents sampled from the C4 corpus (Dodge et al., 2021), focusing on 3 long-tailed factual relations, including {`hasCEO`, `hasSubsidiary`, and `isMemberOf`}. Here, for each SP pair, we need to tap into many thousands of pages, which can be conceptualized as a single long text.

Due to the inherent trade-off between precision and recall, neither metric alone is suitable for our task, and F1 would merely be a generic compromise. The task instead requires maximizing recall, for an effect on knowledge graph (KG) population, with sufficiently high precision to keep downstream curation efforts manageable. Therefore, the metric that we aim to optimize is **Recall@PrecisionX (R@Px)**, where `x` is the minimum precision target to be achieved (e.g., `x` being 50% or, ideally, 80%). In experiments with Llama3.1-instruct-70B (Dubey et al., 2024) as underlying LLM, we reach 80-85%

recall using our passage re-ranking and batching technique in Stage 1, and ~50% R@P50 and ~37% R@P80 through our scrutinization process in Stage 2.

5.1.4 Contributions

The salient contributions of this work are:

1. We introduce the task of extracting a long list of objects for a given subject and relation from long documents (book-length narrative texts).

2. We design and develop the L3X system for this task, combining IR with retrieval-augmented LLM generation and introducing high-recall retrieval, iterative re-ranking with re-prompting, and effective batching.

3. We perform extensive experiments with new benchmarks, showing that L3X outperforms LLM-only baselines that rely on parametric memory, and we analyze strengths and limitations. L3X also outperforms LLM-RAG baselines via effective passage (re)-ranking and batching techniques.

4. We release a publicly accessible system demonstration, `https://d5demos.mpi-inf.mpg.de/l3x`, that supports exploration of multiple L3X configurations, including LLM-only and RAG baselines. The demo helps human knowledge curators understand how the system performs extraction and assists with knowledge-base population.

The dataset, licensing details, code and experimental results are available at `https://github.com/snehasinghania/l3x`.

5.2 Problem Definition

The long document T is either a full book or an aggregation of web pages about a single entity, viewed as a sequence of passages $\{t_i\}_{i=1}^{N}$. Given a subject S, a relation P, and optionally a subset of passages from T, the goal is to generate a complete set $O = \{o_1, o_2, \ldots, o_n\}$ of object entities that stand in relation P with S. The system may additionally produce a confidence score $s(o_i) \in [0, 1]$. The curated benchmarks provide a ground-truth set G of canonical entities with exhaustive alias lists. A prediction o_i is marked correct if, after normalization and disambiguation, it matches any alias of the grouth-truth entity. Alias variants referring to the same entity are canonicalized and deduplicated, and their frequency counts are retained for weighted scoring (detailed explained given in Section 5.3.2).

The learning objective is to maximize recall subject to a minimum precision target $\mathtt{x}$, reported as Recall@Precision$\mathtt{x}$ (R@P$\mathtt{x}$). Let G be the ground-truth set and let $\hat{O}_\tau = \{o : s(o) \geq \tau\}$ be predictions above threshold τ. Then:

$$\text{Precision}(\tau) = \frac{|\hat{O}_\tau \cap G|}{|\hat{O}_\tau|}, \quad \text{Recall}(\tau) = \frac{|\hat{O}_\tau \cap G|}{|G|}$$

$$\text{R@P}\mathtt{x} = \max_\tau \text{Recall}(\tau) \text{ such that Precision}(\tau) \geq \mathtt{x}.$$

Book	**Subject**	**Relation**	**Pred Object**	**score(O)**	**P@Rank**	**R@Rank**
A Song of Ice and Fire	Stannis Baratheon	`opponent`	tywin lannister	18	1.00	0.02
A Song of Ice and Fire	Stannis Baratheon	`opponent`	cersei lannister	14	1.00	0.03
A Song of Ice and Fire	Stannis Baratheon	`opponent`	robb stark	11	1.00	0.05
A Song of Ice and Fire	Stannis Baratheon	`opponent`	roose bolton	11	1.00	0.06
A Song of Ice and Fire	Stannis Baratheon	`opponent`	house lannister	7	0.80	0.06
A Song of Ice and Fire	Stannis Baratheon	`opponent`	boltons	6	0.67	0.06
A Song of Ice and Fire	Stannis Baratheon	`opponent`	eddard stark	6	0.57	0.06
A Song of Ice and Fire	Stannis Baratheon	`opponent`	jaime lannister	6	0.63	0.08
A Song of Ice and Fire	Stannis Baratheon	`opponent`	stannis	6	0.56	0.08
A Song of Ice and Fire	Stannis Baratheon	`opponent`	ironborn	6	0.50	0.08
A Song of Ice and Fire	Stannis Baratheon	`opponent`	salladhor saan	5	0.45	0.08
A Song of Ice and Fire	Stannis Baratheon	`opponent`	paxter redwyne	5	0.50	0.10
A Song of Ice and Fire	Stannis Baratheon	`opponent`	lords tyrell	5	0.46	0.10
A Song of Ice and Fire	Stannis Baratheon	`opponent`	jon snow	5	0.43	0.10

Table 5.1: Example illustrating the metric calculation for R@P50 and R@P80. There are 60 opponents in our dataset for (Stannis Baratheon, `opponent`) SP pair. The correct predictions are marked green and the incorrect ones as red.

Typical targets are $\mathtt{x} \in \{50\%, 80\%\}$. The base cases are: (i) if no τ satisfies Precision(τ) $\geq$ $\mathtt{x}$, we report R@P`x` = 0; (ii) if the generation leads to an empty list and G is not empty (our benchmark doesn't contain no-object cases), then also we report R@P`x` = 0. Moreover, P is drawn from a fixed set of atemporal relations (e.g., `parent`, `friend`, `opponent` for books; `hasSubsidiary` for web). The subject S is a prominent entity in both datasets, and objects O are named entities spanning head and long-tail cases.

Table 5.1 provides an illustrative example of how to compute R@P`x`, where `x` takes two values: 0.5 (50% precision threshold) and 0.8 (80% precision threshold). It shows the generation results for the sample SP pair, (Stannis Baratheon, `opponent`), appearing the *A Song of Ice and Fire* book series, using a single LLM prompt. The predicted objects are sorted in descending order according to the `score`(O) formulation (further details given in Section 5.3.2). To determine the R@P cut-off, we identify the ranks at which precision first meets the threshold (0.5 or 0.8), and then select the corresponding row where recall@rank is highest.

5.3 Methodology

We propose **L3X** for long list extraction from long documents. Figure 5.2 gives an overview of the components, the data flow between them, and the subsequent sections detail each module.

5.3.1 Recall-oriented Generation

The first stage focuses on recall-oriented generation and comprises several components (left panel of Figure 5.2). Given an index over passages of the long text T, a subject S, and a relation P, Stage 1 proceeds as follows:

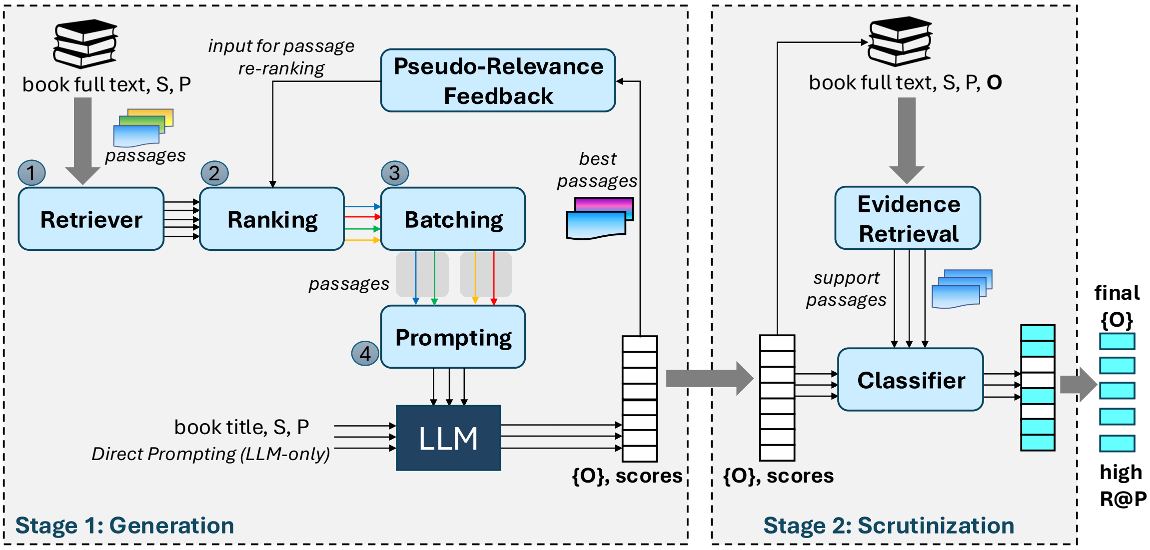

Figure 5.2: Overview of the L3X Methodology.

1. **Retriever:** Retrieve a large set of candidate passages from the long text for the (S, P) pair (e.g., hundreds per query), using the passage index. This is done using a dense retriever (Izacard et al., 2022) by search with S and a set of paraphrases of P (refer Section 5.3.1.1).

2. **Re-ranking:** Rank the passages by various criteria. We present two techniques to prioritize passages based on (i) **num:** number of *named-entity mentions* in a passage, to leverage co-occurrences of multiple O values for the same predicate (e.g., a passage about several friends), and (ii) **amp:** pseudo-relevance feedback (Zhai, 2008) to *amplify* signals from best passages to refine the prompt for the next round (refer Section 5.3.1.2).

3. **Batching:** Batch passages with (i) **neo:** similar entities (including their aliases) identified through named entity overlap, and (ii) **sim:** similar narratives via embeddings, to provide semantically coherent inputs to the LLM (refer Section 5.3.1.3).

4. **Prompting:** LLM is prompted in *retrieval-augmented mode* using the top-ranked passages. The prompt explicitly includes the book title, S and P. For recall, this is an *ensemble* over different choices of retrieved passage (step 1.), and the output of this stage is the union of all objects generated by the LLM (refer Section 5.3.1.4).

The first three steps are optional, enabling simpler configurations. Running only step 4 (without retrieved passages) results in an LLM-only/no-RAG variant, serving as a **direct prompting** baseline. Running only steps 1 and 4 produces a simplified variant of L3X, referred to as **def** (for default configuration), where passages are ranked by retriever scores and batched in the same order. Each of these steps is elaborated below. As Figure 5.2 shows, some of the steps can also be iterated; Section 5.3.1.2 discusses this for the *amp* technique.

5.3.1.1 Passage Retrieval

Long texts, like entire books or web pages combined, are chunked into short passages of 15 sentences, totaling up to 1000 characters. We create all overlapping passages (i.e., with shared sentences) to ensure that sentences with co-references stay connected to named entities in their proximity. Since books often contain extended direct speech, which may omit explicit speaker names, we enrich each passage with *mentions of people and locations* from the *preceding* 10 passages. This metadata annotation ensures that relevant named entity information from prior chunks remains accessible within the current passage.

On the large pool of enriched passages, indexed for efficient retrieval, we select the open-sourced and effective dense retriever, Contriever (Izacard et al., 2022), a BERT-based dense neural IR method fine-tuned on MS-MARCO dataset[1]. The query vector is constructed from the SP pair; an example is: “opponents of Michael Corleone.” Moreover, paraphrases of P and alias names are included, such as “opponents rivals Don Michael” for ensemble mode (see Section 5.3.1.4). Appendix A.2 gives details on query templates.

5.3.1.2 Passage Ranking

The default passage ranking is directly derived from retriever scores. In addition, we introduce two *re-ranking methods* to improve the coverage and relevance of the top-k passages used for prompting.

Default Ranking (def). For a given SP pair, formulated as a natural language query, e.g., *“Who are the friends of Harry Potter”*, the dense retriever ranks top-d passages (with d being a hyperparameter) based on cosine similarity to the query vector. This is the standard IR step to obtain a high-recall first pass over relevant passages (Zhu et al., 2025).

Entity Mention Frequency (num). *re-ordering passages by frequency of named-entity mentions.* We detect mentions of entities (without disambiguation) of the proper type (usually person, place, or org) using spaCy[2] and a hand-crafted dictionary of alias names for S and O, and paraphrases for P (including both nominal and verbal phrases). Top-m passages (with m being a hyperparameter) with higher counts of mentions are prioritized, as they could potentially yield multiple O candidates.

Amplification (amp). *selecting support passages and re-ranking passages by pseudo-relevance feedback.* For this novel *re-ranking* of passages, we employ the IR principle of *pseudo-relevance feedback* (Zhai, 2008). After extracting object lists from the initially selected passages (refer Section 5.3.1.4), we assess the passage quality based on the number of distinct objects the passage yields. The best s passages (with hyper-parameter s) are assumed to provide good cues about relation P in surface form. The averaged embedding vectors for these *high-yield passages* are the reference against which all retrieved passages are re-ranked. The *amp* technique works in two alternating steps and iterates them as follows:

1. For each SP pair, we consider the previously generated O values and the best s high-yield passages: those from which the LLM could extract the most objects.

[1] https://github.com/facebookresearch/contriever

[2] https://spacy.io/usage/linguistic-features#named-entities

2. All retrieved passages are re-ranked by the retriever's scoring model based on combining the original query (about SP) with the selected high-yield passages. The now highest-ranking passages go into the next round of O extraction.

For scoring, we utilize the retriever for computing cosine similarity of passages to a refined query: a convex combination of the original query embedding and the sum of the top-s support passages' vectors:

$$\mathbf{E}(Q') = \alpha\mathbf{E}(Q) + (1-\alpha)\sum_{i=1}^{s}\mathbf{E}(S_i)$$

with embedding function **E( )** and hyper-parameter α. Algorithm 1 gives pseudo-code for *amp*.

Algorithm 1: Iterative Extraction with Pseudo-Relevance Feedback (amp Method).

```
Input: C: candidate pool of retrieved passages; Q: retriever query in natural language with SP
       mentions; k: max no. of passages for prompting LLM; b: batch size; s: no. of support
       passages; α: feedback weight for query reformulation; E: retriever's embedding function
Output: List of object values O
Initialize: O ← ∅ ; q ← E(Q) ;                              //embedding vector
for i ← 1 to ⌈k/b⌉ do
    K ← Retriever(C, q) ;                                    //ranking C for top-k passages
    p_b ← Batching(K) ;                   //b passages by def or by neo/sim (Section 5.3.1.3)
    O_b ← LLM(p_b) ;                       //extracting objects by prompting LLM with passages p_b
    O ← O ∪ O_b ;
    S_b ← ∅ ;
    foreach passage p ∈ p_b do
        if o ∈ O_b appears in p then
            S_b ← S_b ∪ p ;                  //finding support passages from passages p_b
    S ← Top(S_b) ;                           //selecting top-s passages by #objects
    q ← α · q + (1 − α) · (1/s) Σ_{p∈S} E(p) ;     //convex combination to rerank C
return O
```

5.3.1.3 Passage Batching

To feed passages into the LLM, the default approach in the RAG mode is to combine successive ranks into small batches, as determined by the (re-)ranker. Instead, we group the passages into smaller batches of size b (a hyper-parameter; typical values being 2, 4, or 6), based on coherent story structures to aid the LLM in extracting O values (Fan et al., 2024). We devise two criteria for this purpose and batch:

1. **Named Entity Overlap (neo):** passages with a large overlap in named entity mentions;

2. **Passage Similarity (sim):** passages whose embeddings have a high cosine similarity.

For *neo*, we compute Jaccard similarity using min-hash sketches of entity sets, while *sim* uses embedding vectors computed by the retriever. Both strategies process a priority queue of passages as follows: for each rank r (starting with highest, r=1), find the b-1 most related passages from lower ranks (r'>r) to form a batch and prompt the LLM. Mark all the batch passages as "done" and proceed with the next lower rank (r'>r), which is not yet "done".

5.3.1.4 Prompt-based Object Generation

The retrieved top-k passages mention the subject in some form (e.g., first name, last name, or alias) and may contain other named entities. We append the passages into the prompt context for retrieval-augmented list generation (Liu et al., 2023a; Gao et al., 2023; Zhao et al., 2023). As LLMs have limits on input context (and GPU memory demands increase with input length), we divide the top-k passages (ranked by retriever scores) into batches of b passages each (e.g., k=20, b=4 gives 5 batches). The O values generated from batch-wise processing are combined by their union for high recall.

Prompts can optionally include a small set of demonstration examples for in-context inference. These examples explicitly mention SP appearing in books disjoint from the dataset, along with their complete O lists, aiding the instruction-tuned LLM in object list generation. We refer to this mode as **few-shot** prompting, while the basic mode without examples is referred to as **zero-shot**. Table 5.2 shows an example for the few-shot prompt formulations, and Appendix A.2 in this chapter gives complete details on the prompt design for each relation.

System:- You are a knowledge base. Generate the complete list of names (Objects) who are parents, including step parents, of the specified person in the given book. List the names one after the other, separated by commas.
Few-Shot examples: Input: Book: A Promised Land, Subject: Barack Obama, Relation: parent Output: [Barack Obama Senior, Stanley Ann Dunham] Input: Book: The Fellowship of the Ring, Subject: Frodo Baggins, Relation: parent Output: [Drogo Baggins, Primula Brandybuck]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation:{P}, Passages: {T}

Table 5.2: Example of prompt template for `parent` relation (placeholders in curly brackets).

In **single-prompt** mode, the LLM uses only the best of these formulations (i.e., considered most natural by humans). In **ensemble** mode, for each relation, we manually prepare five prompt templates for direct prompting, and five retriever query templates for L3X-RAG, and repeat all LLM-based extraction tasks with all templates. The final O is the union of the O values generated across all runs. Of the four configurations (zero-single, zero-ensemble, few-single, few-ensemble), we report main results for the **few-ensemble** setting, with the other configurations evaluated in ablation studies (refer Section 5.6.5).

5.3.2 Precision-oriented Scrutinization

In the precision-oriented scrutinization stage, we leverage the fact that, unlike in the first stage, we now have lists of candidate objects. To scrutinize the candidate objects O for a given SP and eliminate false positives, we devise several techniques. The key idea is to identify passages that clearly reflect SPO triples, and use these *support passages* to rank and prune O values, and also learn embeddings for the P predicates. Figure 5.2 (right panel) gives an overview.

Scoring of O Candidates. Each LLM call returns a list with a score for the entire list, no scores for individual objects. However, with batch-wise LLM calls and the ensemble with different prompts, we can derive a total score for each O candidate (for a given SP), by a weighted occurrence frequency:

$$\texttt{score}(O) = \sum_{\text{batch}_i} \exp\left(\text{score}_{\text{LLM}}(L_i)\right) \times \mathbf{I}_i(O)$$

where $\mathbf{I}_i(O)$ is an indicator variable set to 1 if O occurs in the output list L_i for the ith batch of passages, and zero otherwise. $\text{score}_{\text{LLM}}$ is the log probability given by the language model. This can then be used for direct pruning by thresholding on scores.

5.3.2.1 Evidence Retrieval

While stage 1 needs to start the retrieval with S and P only, stage 2 has O candidates at its disposal. This allows us to search the entire book for textual snippets that explicitly indicate SPO triples. For each SPO candidate, we retrieve the top-s passages, termed the *support passages* for SPO. These are different from the high-yield passages used by the *amp* method in stage 1, as we now retrieve from scratch from the entire book.

For retrieval, we generate passage embeddings using the retriever's text-to-vector model. The vectors are compared against embeddings of the concatenated SPO strings, including SO alias names and paraphrases of P, using cosine similarity.

5.3.2.2 Classifier: Score-based Thresholding (thr)

The simplest scrutinizing technique is to prune O candidates below a a specified cut-off point in the ranked list of per-O scores ($\texttt{score}(O)$). As the score distribution is often heavily skewed, we do not truncate by score value, but set the cut-off point to be the tth quantile of the cumulative score distribution, with the default setting t=0.8. That is, we keep only the highest-ranking O values that constitute 80% of the score mass.

5.3.2.3 Classifier: Confidence Elicitation (conf)

We prompt the LLM again to assess its confidence in the generated O values. For each *SPO*, top-p support passages in their enriched form (with all named entities including S and O) are included into the LLM prompt for in-context inference: "Given this information, is *SPO* a correct statement?". The *conf* classifier accepts an O candidate if the LLM gives a "yes" reply. This approach differs from the passage-based extraction of the recall-oriented stage, as the support passages are retrieved individually for each O-candidate.

5.3.2.4 Classifier: Predicate-specific Classifier (pred)

The collection of support passages, for all SO with the same predicate P, can be utilized to learn an embedding for P cues, sort of a "mini-LM" for P. The intuition is that support passages with indicative phrases, such as "life-or-death combat with", "deeply hates" or "I will destroy you" (in direct speech), can collectively encode a better signal for P. To construct the classifier, we perform the following steps:

1. For each O, we retrieve top-p support passages, and encode them into embedding vectors.
2. We identify the top-ranked O values with $\texttt{score}(O)$ above a threshold ω.
3. Using the top-ranked O, we combine the per-O passage vectors by a weighted sum, with $\texttt{score}(O)$ as weights, to obtain a single P-vector (classifier).
4. Each SO pair under scrutiny (O below the threshold ω) is tested by comparing the vector of the top-p support passages for this SPO candidate against the P-vector computed using steps 1 to 3.
5. The classifier accepts a low-ranked SO if the cosine similarity between the embeddings is above a threshold θ.

We construct a *pred* classifier for each SP pair, in a completely self-supervised manner. It has hyper-parameters ω, p and θ, though; these are tuned via withheld train/dev data with SPO ground-truth, but without any supervised passage labels.

5.3.2.5 Classifier: Discriminative Classifier (dis)

Another way of harnessing the SPO support passages is to train a discriminative classifier, again in a self-supervised manner. We consider the ranked list of O values for a given SP and pick:

- the top-q high-scoring O candidates
- the bottom-r low-scoring O candidates

with q and r as hyper-parameters. For each top-q and bottom-r candidate O, we retrieve top-p support passages, forming one passage pool for the high-scoring Os and another pool for the low-scoring Os. In each of these pools, the passages are cast into embeddings, and weighted averaged with $\texttt{score}(O)$ to form SP_{high} and SP_{low} vectors.

Finally, each candidate O for a given SP is classified by whether its own support-passage vector is closer to the SP_{high} or the SP_{low} vector, in terms of cosine distance, leading to acceptance or rejection, respectively.

5.4 Experimental Setup

5.4.1 Datasets

We make use of *fiction books* as a most representative, primary target for experimental studies. A second dataset, on *web contents with business-oriented relations*, exhibits different characteristics, and adds diversity to the experiments. The results go into more depth and variety on the books data (Section 5.4.1.1), and are shorter on the complementary web data (Section 5.4.1.2).

5.4.1.1 Books Dataset

The task of extracting long O lists from long texts is novel, with no suitable benchmark datasets available. Therefore, we constructed a new dataset of books and corresponding ground-truth O lists associated with SP pairs. We selected eleven popular novels and entire book series, including [*A Song of Ice and Fire Series, Godfather Series, Harry Potter Series, Outlander Series, Little Women, Malibu Rising, Pride and Prejudice, Steve Jobs, The Girl with the Dragon Tattoo, Wuthering Heights, The Void Trilogy*], enthusiastically discussed on community websites[3]. These fan communities feature extensive lists and infoboxes from which we derived SPO ground-truth with high confidence. As detailed in Section 5.3.1.1, the total no. of passages per book varies, ranging from ca. 10,000 passages in epic book series like *A Song of Ice and Fire* to ca. 700 passages in shorter books like *Malibu Rising*.

Since entities often appear under multiple surface forms, we manually constructed an entity name dictionary grouping alias names for each distinct entity. On a per-book basis, we ensured that certain first names, last names, or nicknames were uniquely identifiable. For example, "Daenerys" is unique, whereas "Targaryen" is ambiguous. So for this entity, aliases include "Daenerys", "Dany", "Daenerys Targaryen", "Daenerys Stormborn", but not "Targaryen". LLM outputs like "Targaryen" alone are thus counted as false. This construction was aided by additional community sources[4].

Our books dataset comprises 764 distinct SP pairs for 8 predicates. In total, it covers ~5300 entities that appear under ~12,000 alias names. While the S entities are prominent book characters, their associated O lists are long and dominated by rarely mentioned, long-tail entities. To highlight the gap with standard RE, we examined the Wikidata knowledge graph (KG) (Vrandecic, 2012) for triples involving the 30 Harry Potter characters used as target S. While the KG includes most of our predicates, it lacks substance beyond metadata (e.g., featured-in-media, library IDs). It is also extremely sparse: for instance, it lists only 2 of Harry's enemies, compared to 50+ in our ground truth—a trend consistent across other subjects and predicates.

Relation Difficulty. The chosen 8 predicates include 3 *easy relations* with a limited number of O values (`parent`, `child`, and `sibling`) and 5 *hard relations* with potentially long O lists (`family`, `friend`, `opponent`, `placeHasPerson` (i.e., people being at a place), and `hasMember` (i.e., members of organizations or events)). For instance, the longest O list for the `opponent` relation (i.e., enemies, rivals etc.) has 60 distinct entities; the average length is 9 with a high standard deviation of 11. Table 5.3 in gives books dataset statistics.

5.4.1.2 Web Dataset

To demonstrate the generalizability of L3X, we constructed a second dataset with partly similar and partly complementary characteristics. The data is derived from the large common crawl of web pages (C4 corpus) (Dodge et al., 2021), and we aim to extract long object lists for three business/biography relations: CEOs of companies (including past ones), subsidiaries of companies, and organizations

[3] `www.cliffsnotes.com`, `www.bookcompanion.com`, `www.fandom.com`

[4] including `www.potterdb.com` for Harry Potter, `www.reddit.com/r/asoiaf` for Song of Ice and Fire, and others

Relation	Type	#Subject	#Object per Subject	
			range	μ (σ)
`parent`	person→person	85	1–4	1.9 (0.6)
`child`	person→person	48	1–9	3.3 (2.4)
`sibling`	person→person	65	1–8	3.0 (1.8)
`family`	person→person	81	1–47	12.1 (9.8)
`friend`	person→person	99	1–85	11.1 (16.5)
`opponent`	person→person	88	1–60	8.9 (11.2)
`placeHasPerson`	location→person	189	1–92	6.7 (12.6)
`hasMember`	organization→person	109	1–142	11.6 (20.5)

Table 5.3: Books Dataset Statistics.

Relation	Type	#Subject	#Object per Subject	
			range	μ (σ)
`hasCEO`	organization→person	100	2-43	8.5 (5.2)
`isMemberOf`	person→organization	100	3-74	20.1 (14.7)
`hasSubsidiary`	organization→organization	100	2-308	71.8 (52.0)

Table 5.4: Web Dataset Statistics.

that a famous person is part of (e.g., companies, societies, charities, schools at different levels). The dataset has 100 subjects for each P, covering ~6400 entities appearing under ~24,000 alias names. Table 5.4 gives per-relation statistics.

5.4.2 System Configurations

The presented L3X methodology comes with many options for its different components, with the most important choices occurring for ranking, batching and scrutinization. In our experiments, we focus on configurations for these three components, labeling them accordingly, e.g., as *num/sim/thr* or *amp/neo/pred*. The *thr* classifier with default setting $t = 0.8$ is abbreviated as *thr80*.

5.4.2.1 Hyper-Parameters

L3X comes with tunable hyper-parameters. Table 5.5 lists the default values for most experiments. We widely varied these values, reporting notable cases in Section 5.6.5 on sensitivity studies.

The best settings were identified using withheld train/dev data. To this end, we split the datasets into three folds (30:20:50), through stratified sampling on books and SP pairs, ensuring equal representation of varying O-list lengths in both folds. For each subject in train/dev, the complete O list is taken from the ground truth, to prevent information leakage into the test set. The best hyper-parameter values are determined through grid search, maximizing the recall metric in Stage 1 and the R@P50 metric in Stage 2 (or alternatively AUC). This is done for two modes: a single *global* value for a hyper-parameter, or *per-predicate* values, specific for each P.

Parameter	Tuned Values
l: passage length (#char)	1000
passage overlap (#char)	200
preceding passages for annotation	10
d: # retrieved passages	500
k: top-k passages	40
b: # passages per batch	2
m: # passages in *num*	50
s: top-s high-yield passages in *amp*	2
α: feedback weight for *amp*	0.7
t: percentile retained for *thr*	0.8
p: top-p support passages for *conf*	2
ω: cut-off score for O values in *pred*	50
p: top-p support passages for *pred* and *dis*	5
θ: acceptance bound for *pred*	0.85
q: top-q O candidates for *dis*	50
r: bottom-r O candidates for *dis*	50

Table 5.5: Hyper-Parameters for L3X

5.4.2.2 Evaluation Metrics

By the design rationale of L3X, we use different metrics for stage 1 and stage 2. For the recall-oriented stage 1, the obvious measure of interest is *recall*: the fraction of ground-truth object (O) values correctly generated. For stage 2, neither precision nor recall alone reflect our objective, and F1 would merely be a generic compromise. Instead, we aim to achieve high recall while keeping precision at an acceptable level. Therefore, our key metric—computed from the final ranked lists—is **Recall@PrecisionX (R@Px)**, where x is the precision to be guaranteed (e.g., x being 50% or, ideally, 80%). R@Px metric reflects the need for high-coverage outputs worthwhile for downstream applications such as tool-supported literature analysis, while avoiding too many errors as these entail manual curation. For both stages, we also report *precision* values and the *precision-recall area under the curve* (AUC).

Moreover, all reported numbers are *macro-averaged percentage* scores, computed in three steps. For each SP pair, we first compute the precision and recall of the generated O list against the ground-truth. These scores are then averaged across all SP pairs for each P. Finally, the results are averaged across all relations.

5.5 Results on the Books Dataset

The main findings on the long list generation task on the primary, books, dataset are given below.

LLM	Config	Stage 1			Stage 2 (thr, t=0.8)				
		P	R	AUC	P	R	AUC	R@P50	R@P80
GPT-3.5	zero-single	41.1	39.2	16.7	37.6	25.9	13.0	19.4	14.2
	zero-ensemble	32.0	49.8	20.6	32.4	41.5	18.8	30.0	21.7
	few-single	**46.8**	38.1	19.7	**43.9**	25.6	16.2	20.3	16.3
	few-ensemble	43.6	43.9	21.3	41.5	31.4	17.4	25.4	20.2
Llama-8B	zero-single	15.3	28.9	6.7	14.7	23.5	5.5	11.5	6.2
	zero-ensemble	8.4	39.0	10.0	9.3	39.0	9.3	18.4	9.6
	few-single	26.6	26.4	9.3	23.8	17.8	7.9	11.7	8.7
	few-ensemble	21.2	31.9	11.5	21.2	27.8	10.5	16.9	12.4
Llama-70B	zero-single	41.9	38.7	15.8	42.6	28.6	15.0	19.6	15.9
	zero-ensemble	32.1	49.6	20.2	34.2	41.9	19.8	29.4	21.3
	few-single	38.5	43.4	18.2	39.1	31.9	16.9	23.7	17.6
	few-ensemble	34.1	47.7	20.5	37.0	39.5	20.5	31.4	21.9
	RAG-single	16.9	78.0	18.8	19.1	72.0	20.9	36.5	22.8
	RAG-ensemble	12.0	**84.3**	**22.9**	14.6	**82.8**	**22.7**	**40.2**	**26.1**
	Oracle	-	87.8	-	-	-	-	-	-

Table 5.6: Complete Results on LLM-only and RAG Baselines.

5.5.1 Performance of LLMs and RAG

Table 5.6 reports macro-averaged results for the LLM-only setting with three widely used models, including GPT-3.5[5], Llama3.1-8B[6], and Llama3.1-70B[7]. The LLMs are prompted using various direct prompting techniques as described in Section 5.3.1.4, and all use *thr80* (t=0.8) for stage 2.

We notice that LLM-only performance is poor, achieving less than 50% recall after stage 1, with mediocre precision. Llama-70B and GPT-3.5 perform comparably, while Llama-8B substantially lags behind. Since Llama-70B has the best performance in the LLM-only in terms of recall (47.7%), we also present Llama-70B results used in the RAG mode.

In the RAG mode (with *def* ranking of passages), results improve: 84% recall after stage 1, but precision stays low even after *thr80*-based scrutinization. The best R@P50 number is 40.2%. As a reference, we estimate an oracle upper bound of 88% by counting the distinct O values from ground truth that appear in at least one of the retrieved top-500 passages. The insight here is that LLMs can recall only a fraction of O's from pre-training, and add many false positives. Equipped with book passages, the recall is improved, but false positives remain a major challenge for this very difficult task.

[5]platform.openai.com/docs/models/gpt-3.5-turbo
[6]https://huggingface.co/meta-llama/Llama-3.1-8B
[7]https://huggingface.co/meta-llama/Llama-3.1-70B

Config		Stage 1			Stage 2 (thr80)				
		P	R	AUC	P	R	AUC	R@P50	R@P80
def ranking	RAG	12.0	84.3	22.9	14.6	82.8	22.7	40.2	26.1
reranking	num	11.8	**85.2**	21.8	14.8	**83.1**	22.7	39.7	24.8
	amp	13.7	83.6	**27.5**	16.0	81.0	**27.4**	**48.6**	**35.9**
batching	neo	12.6	83.8	22.4	15.5	81.6	23.0	39.1	25.1
	sim	12.5	84.2	22.2	15.3	82.5	23.2	39.4	23.5
reranking+batching	num+neo	12.9	85.0	22.3	15.2	81.9	22.8	38.0	25.0
	amp+neo	**14.1**	83.4	27.1	**16.7**	81.6	26.9	47.7	35.4
	num+sim	12.1	83.8	21.8	14.7	81.5	22.4	38.0	24.8
	amp+sim	**14.1**	83.4	27.1	16.0	80.5	26.3	47.0	33.8

Table 5.7: Results for Stage 1 configurations, with few-ensemble prompting and Llama-70B model, on Books Data (ranking:{num, amp}; batching:{neo, sim}; top-k=40; default thresholding ($t = 0.8$) for Stage 2).

5.5.2 Added Value of L3X Configurations

Table 5.7 compares different L3X configurations, contrasting them with Llama-70B model in RAG mode. All L3X configurations greatly improve recall, up to almost 85% and adding smart re-ranking (*amp*) and batching to RAG pays off very well. After stage 1, the recall by L3X variants is similar to the RAG baseline, but the best option for recall is the *num* method with entity-count-based re-ranking. However, in terms of AUC, the method that shines most is the iterative *amp*, leading to notable improvement with AUC value of 27.4%. This indicates a higher concentration of true positives among the top-ranked O values—an important asset for stage 2.

The AUC of *amp* is a strong starting point for the *thr80* classifier at Stage 2, where it achieves the best R@P values, with 48.6% for R@P50—with a large margin over the second-best method.

5.5.3 Boosting Recall While Maintaining Precision

We pick the best performing stage 1 configurations, namely, *amp* and *amp+neo*, plus the default RAG *def* for contrast. Table 5.8 shows the stage 2 results with different classifiers for scrutinization. The key findings are highlighted below.

Best Configurations pred and dis. The *thr80* technique already works fairly well, especially with tuning of its hyper-parameter t. The more sophisticated classifiers *pred* and *dis* still have an added benefit, improving the final R@P values up to 49.7% for R@P50. The bottom line is that L3X *amp* adds substantial benefits over LLM-only and standard RAG methods, highlighting the crucial role of judicious passage ranking. The final R@P results—reflecting the benefit/cost ratio for downstream usage—are promising, but still fall short of being fully satisfying. This emphasizes the challenging nature of the new task explored in this chapter.

	Config	P	AUC	R@P50	R@P80	R@P90
RAG	thr-g	14.6	22.7	40.2	26.1	24.1
	thr-p	16.5	22.1	39.5	25.9	23.9
	conf	43.6	18.2	28.4	18.4	17.8
	pred-g	18.9	23.4	41.3	26.6	24.8
	pred-p	21.5	22.9	40.7	26.5	24.7
	dis-g	20.2	23.1	41.2	26.6	24.8
	dis-p	21.4	21.8	40.1	26.2	24.7
amp	thr-g	16.4	27.3	48.5	35.8	32.9
	thr-p	17.5	27.2	48.5	35.8	32.9
	conf	**46.6**	19.0	31.7	20.4	19.3
	pred-g	20.4	**28.1**	**49.7**	**36.5**	**33.3**
	pred-p	23.5	28.0	48.7	36.2	33.0
	dis-g	21.5	27.9	**49.7**	**36.5**	**33.3**
	dis-p	21.2	27.5	49.6	36.4	**33.3**
amp+neo	thr-g	16.4	26.9	47.7	35.4	33.1
	thr-p	17.4	26.7	47.5	35.3	33.0
	conf	45.4	18.7	30.7	20.1	19.6
	pred-g	19.8	27.6	48.7	35.7	33.1
	pred-p	22.1	27.4	48.0	35.4	32.9
	dis-g	20.7	27.4	48.7	35.7	33.1
	dis-p	20.9	26.2	48.0	35.4	32.8

Table 5.8: Results for L3X Stage 2 Configurations (classifiers: {thr, conf, pred, dis}, g refers to global grid search and p is per-predicate grid search).

Hyper-parameter Tuning for pred. The *pred* method has three hyper-parameters. Setting their values by global grid search with train/dev data leads to the best results, with ω=20, p=5, and θ=0.75. As the various P exhibit different characteristics, we would expected even further gains with the per-predicate grid search. Indeed, this led to rather different predicate-specifc values. For example, for `sibling`, the best values are ω=10, p=2, θ=0.9, whereas for `friend` we have ω=50, p=1, θ=0.55. This makes sense, as `sibling` lists are much shorter, and long `friend` lists have much noisier support passages. So the setting is tighter for `sibling`, and has more liberal θ for `friend` but only 1 support passage per O to tame noise. Nevertheless, the *pred-p* and *dis-p* techniques did not achieve significant improvements over the globally tuned variants *pred-g* and *dis-g*. We attribute this result to the fact that the simpler configurations are already close to the best possible outputs

Relation	RAG	L3X-amp	L3X-amp-neo
parent	75.6	76.2	73.8
children	86.5	82.5	84.6
sibling	87.2	86.2	89.3
avg. Easy P	83.1	81.6	82.6
family	79.8	79.8	78.5
friend	85.4	85.5	85.2
opponent	80.8	81.1	80.1
hasMember	89.0	86.6	86.1
placeHasPer	89.8	90.7	89.4
avg. Hard P	85.0	84.7	83.9
avg. All P	**84.3**	83.6	83.4

Table 5.9: Drill-Down Recall Results by Predicate after Stage 1.

given the inherent difficulty of the task.

Confidence Elicitation from LLM. *conf* performs poorly achieving 46.6% precision, 44.7% recall, 19.0% AUC, 31.7% R@P50 and 20.4% R@P80 with the L3X *amp* configuration, as the LLM becomes rather conservative when fed with support passages about SPO candidates (Wang et al., 2023a), rejecting too many valid entries.

5.6 Analysis and Discussion

5.6.1 Drill-down by Predicate

The reported results are macro-averaged over all relations. However, some relations P are easier to deal with than others (see Section 5.4.1.1). We analyzed performance per predicate using the best configurations after stage 1 in Table 5.9 and stage 2 in Table 5.10.

Stage 1 recall is fairly consistent across predicates (75-90%), but stage 2 R@P numbers vary widely:"easy" relations with short, well-defined lists perform well, while "hard" relations—those with longer lists and vaguer cues—show a significant drop. As expected, `opponent` is the most difficult predicate, where even our best method reaches only 32.5% of R@P50. This calls for more research on this challenging task.

5.6.2 Influence of Batching

When combined with batching via *neo* or *sim*, L3X improves in precision, but loses recall, and eventually stays inferior to *amp* alone. While unexpected, it is not counter-intuitive: *amp* operates iteratively, and judiciously picks its batch of passages in each round. So batching does help, but the big mileage already comes from the amplification of support passages. However, for specific predicates there are gains from the *neo* batcher in Stage 1. Table 5.9 presents the drill-down by

Relation	RAG-pred(g)			L3X-amp-pred(g)			L3X-amp-neo-pred(g)		
	AUC	R@P50	R@P80	AUC	R@P50	R@P80	AUC	R@P50	R@P80
parent	27.3	57.7	47.0	27.9	61.3	52.4	27.9	61.3	53.6
children	28.1	60.9	43.8	36.5	72.3	61.0	34.4	66.1	55.2
sibling	38.0	65.4	50.4	47.7	79.4	67.7	46.3	77.9	66.0
avg. Easy P	31.1	61.3	47.1	37.3	71.0	60.3	36.2	68.4	58.3
family	25.2	34.0	15.0	33.1	44.8	33.2	32.9	46.2	32.1
friend	19.7	27.1	13.1	23.8	35.5	17.0	23.2	35.5	18.6
opponent	17.6	29.3	14.5	18.9	32.4	14.6	18.8	30.9	14.2
hasMember	16.5	25.7	14.0	20.7	32.5	20.9	21.1	36.5	20.9
placeHasPer	14.7	30.3	15.3	16.5	39.6	25.0	16.1	35.6	24.8
avg. Hard P	18.7	29.3	14.4	22.6	37.0	22.1	22.4	36.9	22.1
avg. All P	23.4	41.3	26.6	**28.1**	**49.7**	**36.5**	27.6	48.7	35.7

Table 5.10: Drill-Down Results by Predicate after Stage 2.

predicate results to showcase this results. Replacing the *thr* pruning with the sophisticated *pred* classifier further enhances the performance a bit. Again, in Table 5.10 drill-down by predicate shows higher gains for some of the hard P (`family`, `friend` and `hasMember`), indicating potential for more.

5.6.3 Influence of Entity Popularity

We further analyzed performance by splitting ground-truth O entities in the test set into *head* and *tail* groups, based on their frequency in the book. Entities above the 75th percentile were labeled as head, the rest as tail. This results in four combinations: (easy P, head), (easy P, tail), (hard P, head), and (hard P, tail). Table 5.11 reports results on these four combinations. We observe that *amp* consistently outperforms standard RAG across all four cases. However, in the most challenging setting—hard P with tail O—performance drops sharply.

5.6.4 Book-specific Performance

Most books in our data are well discussed in online media (including movie/TV adaptations). The LLM implicitly taps into this contents by its parametric memory. To assess the influence of pre-training, we compare the performance of various configurations on a per-book basis.

Tables 5.12 and 5.13 reports per-book results, comparing LLM-only, standard RAG and our L3X-amp method, all with *thr80* for Stage 2 pruning. It is evident that extraction/generation quality correlates with the popularity of the books and the frequency of online content *about* them (e.g., in discussion forums and social media). This trend is most pronounced for *Harry Potter* and *A Song of Ice and Fire*. The book with the weakest results is the *Void trilogy*, whose ground truth comprises 498 SPO triples for 57 unique subjects. Although it is popular among science fiction fans, there is much less online content about it, and it appears that it is not among the pre-training data of

Config		Stage 1			Stage 2 (thr, t=0.8)				
		P	R	AUC	P	R	AUC	R@50	R@80
rag	EH	18.0	90.3	26.3	22.2	89.3	26.4	65.9	46.1
	ET	16.7	79.3	15.3	19.9	74.9	16.8	35.7	26.0
amp	EH	23.4	87.2	32.1	26.1	83.0	31.5	74.0	55.1
	ET	17.3	75.9	18.9	21.1	73.1	20.2	45.5	35.0
rag	HH	2.4	92.1	17.3	3.0	91.6	17.4	32.4	16.6
	HT	3.0	75.3	8.5	3.6	73.7	8.3	8.0	3.1
amp	HH	2.5	91.9	20.6	3.2	91.1	20.4	40.5	24.7
	HT	3.1	74.9	10.4	3.5	72.1	10.4	13.1	4.1

Table 5.11: Results on Head-vs-Tail entities. EH: easy P+head, ET: easy P+tail, HH: hard P+head and HT: head P+tail.

LLMs. Notably, though, the L3X-amp method significantly improves on all metrics, compared to both LLM-only and RAG. For example, R@P80 improves from less than 10% to 28%.

Book	**P**	**AUC**	**R@P50**	**R@P80**
A Song of Ice and Fire	45.6	25.1	38.4	26.3
Godfather	30.4	12.2	16.6	11.3
Harry Potter	50.8	25.8	36.8	27.9
Little Women	21.2	9.0	24.8	9.4
Malibu Rising	15.7	11.2	20.1	15.9
Outlander	23.2	10.2	21.0	13.8
Pride and Prejudice	32.7	19.8	30.3	22.1
Steve Jobs	43.4	18.9	32.3	24.0
Girl with the Dragon Tattoo	19.6	10.9	15.7	14.3
Void Trilogy	8.2	6.6	7.4	7.3
Wuthering Heights	50.5	35.3	37.6	37.6

Table 5.12: Drill-Down Results by Book using LLM-only and *thr* ($t = 0.8$) pruning for Stage 2.

5.6.5 Sensitivity of Hyper-Parameters

We performed extensive experiments on varying hyper-parameter settings. We report on the sensitivity of the two most important Stage 1 choices: k (number of top-k passages) and b (number of passages per batch). Figure 5.3 shows *R@P50* numbers for *amp/neo* by varying k from 5 to 40 and b from 2 to 5.

Book	RAG				L3X-amp			
	P	AUC	R@P50	R@P80	P	AUC	R@P50	R@P80
A Song of Ice and Fire	17.3	29.2	42.9	31.0	17.3	37.8	55.7	41.8
Godfather	9.4	17.9	31.2	15.0	9.9	18.0	36.3	25.3
Harry Potter	18.2	33.9	49.4	31.8	28.8	37.0	52.2	38.9
Little Women	15.6	14.5	26.3	15.6	14.0	15.7	28.6	18.2
Malibu Rising	10.5	13.0	18.7	14.1	10.8	16.8	21.9	21.9
Outlander	3.7	10.7	23.6	16.3	4.4	13.3	30.6	15.5
Pride and Prejudice	17.7	18.2	34.7	20.4	18.5	22.8	39.0	22.9
Steve Jobs	11.7	31.9	57.0	36.5	12.0	32.6	55.6	41.8
Girl with the Dragon Tattoo	7.2	16.2	27.0	4.5	8.5	17.3	29.7	26.5
Void Trilogy	2.5	6.4	10.3	8.5	2.5	8.3	34.1	28.0
Wuthering Heights	35.5	24.7	45.4	39.7	34.1	26.2	44.1	38.4

Table 5.13: Drill-Down Results by Book using RAG and L3X-amp, both with *thr* ($t = 0.8$) pruning for Stage 2.

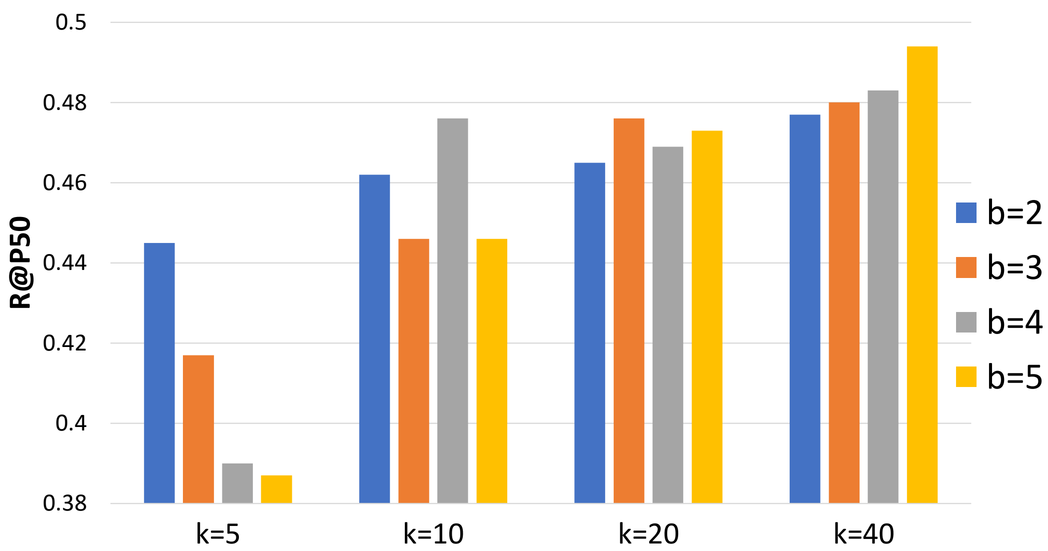

Figure 5.3: Varying hyper-parameters k, b for *L3X-amp-neo*.

amp	Stage 1			Stage 2 (thr, t=0.8)				
	P	R	AUC	P	R	AUC	R@P50	R@P80
#q=5	13.7	83.6	27.5	16.0	81.0	27.4	48.6	35.9
#q=4	14.2	82.5	27.3	16.6	79.8	27.0	48.0	36.0
#q=3	15.2	81.1	27.2	18.1	78.5	27.5	48.2	35.4
#q=2	17.2	79.9	27.2	20.1	76.8	27.2	47.2	34.8
#q=1	20.5	76.2	26.1	23.4	71.1	26.5	44.0	31.5

Table 5.14: Varying no. queries for retriever with *amp*.

We observe that increasing k is very useful, but has diminishing returns beyond a certain value. Note that larger k also increase the cost for invoking the LLM more often. For small k, there is no benefit from larger batches, but this changes for large k, where bigger batches keep improving recall. Moreover, this also increases LLM costs, as we pass more tokens into the inference.

All L3X configs use five reformulations for retriever queries. Table 5.14 presents the change in results with fewer query variants using *amp* configuration. Both recall and R@P values drop with less queries, showing the vital role of diversified formulations.

5.6.6 Ground-Truth Variants

As detailed in Section 5.4.1, the ground-truth O lists are derived from online sources with high quality control. Inevitably, such lists are often incomplete, particularly for books with hundreds of minor characters. Therefore, we also evaluate by *pooling-based ground truth*, where the true positives within the union of all O values returned by all the methods is the complete ground truth.

In this evaluation mode, stage 1 recall by *def* and *amp* increases by ~10 points, reaching 93% and 92%, respectively. This is intuitive, as pooled ground truth is a subset of the fully hand-crafted ground truth. The numerical gains carry over to Stage 2: with *thr80*, *amp* goes up to 56.5% R@P50 and 41.4% R@P80.

5.6.7 Other Evaluation Metrics

With micro-averaging rather than macro-averages, the relative gains/losses across configs hardly change. The best results are still with *amp*, reaching 84.7% recall and 24.9% AUC in Stage 1, and 44.7% R@P50 and 31.6% R@P80 in Stage 2 and *thr80*.

Since the goal is to extract long lists, Precision@k and NDCG@k at various ranks can also be easily computed. For the most notable configurations and k values, the numbers are reported in Table 5.15. These numbers show that LLM-only starts out with better precision, but it generates much shorter lists, thus falling short on recall. If we were to pad its output lists with negatives, the numbers for $k = 50$ and $k = 100$ would match those at $k = 20$.

For the RAG and amp, on the other hand, we can go deeper into their ranked lists. Precision naturally declines with increasing k as lower ranks have more negatives. The NDCG values, though, exhibit the benefits of L3X: this metric reflects the distribution of positive items in the

Config	p@10	p@20	p@50	p@100	ndcg@10	ndcg@20	ndcg@50	ndcg@100
LLM-only	0.34	0.31	-	-	0.43	0.42	-	-
RAG	0.26	0.21	0.15	0.12	0.48	0.50	0.53	0.54
amp	0.30	0.23	0.17	0.14	0.56	0.57	0.60	0.61

Table 5.15: Results with Precision@k and NDCG@k metrics. '—' is not applicable due to generated lists being shorter than the value for k.

ranking, emphasizing whether positives are concentrated near the top or dispersed toward the bottom. In this respect, L3X continues to improve its NDCG scores even up to $k = 100$. Beyond this point (for SP pairs with even longer O lists), the improvement stagnates and eventually declines slightly due to positives appearing at very low ranks.

5.6.8 Comparison with Other Related Work

5.6.8.1 Relational Information Extraction

Since Stage 2 processes full SPO triples, we can apply standard relational IE for classifying whether an SO pair satisfies a given relation P. We considered two state-of-the-art methods, GenIE (Josifoski et al., 2022) and DREEAM (Ma et al., 2023b), and fine-tuned them on our predicates with the train/dev fold. However, both these models performed very poorly, with recall below 5% and precision no higher than 10%. Clearly, our task is outside their comfort zone, as passages from fiction books are very different from Wikipedia paragraphs for which these methods were originally designed and trained on. These results highlight the distinctive and challenging nature of the proposed long-lists-from-long-documents task.

5.6.8.2 GraphRAG

For complete performance comparison, we ran GraphRAG (Edge et al., 2025) on our task in the following setup.

1. We tuned the entity and relationship extraction prompt of GraphRAG for our task. This was done by including in-context examples for each predicate in our dataset.

2. We used GPT-3.5 model, which has size comparable to Llama-70B

3. Each book is indexed once for querying on the test dataset

We evaluate the performance using precision, recall, F1, and AUC for Stage 1 (without scrutinization) and for Stage 2 (with scrutinization by our *thr*, $t = 0.8$ pruning method), in the few-shot prompting mode. Table 5.6.8.2 reports the results. We notice that GraphRAG doesnot outperform the other methods, in neither of the two stages. Despite adding expressiveness, GraphRAG does not yield better recall than L3X, nor better precision than GPT-3.5.

Table 5.6.8.2 also gives the performance drill-down by predicate for GraphRAG and L3X-amp to gain a complete comparison. We observe that GraphRAG yields decent precision but lower recall

Config	Stage 1				Stage 2			
	P	**R**	**F1**	**AUC**	**P**	**R**	**F1**	**AUC**
LLM-only (GPT-3.5)	**46.8**	38.1	**42**	19.7	**37.6**	25.9	30.7	13
GraphRAG (GPT-3.5)	30.3	45.5	36.4	12.7	27.5	34.6	30.6	9.1
L3X-RAG (Llama-70B)	16.9	**78.0**	27.8	18.8	19.1	**72.0**	30.2	20.9
L3X-amp (Llama-70B)	20.5	76.2	32.2	**26.1**	23.4	71.1	**35.2**	**26.5**

Table 5.16: Comparison with *GraphRAG* model.

Relation	GraphRAG				L3X-amp			
	P	**R**	**F1**	**AUC**	**P**	**R**	**F1**	**AUC**
children	31.3	51.5	33.0	11.3	30.3	77.8	43.6	35.1
parent	39.8	82.7	50.9	24.5	42.0	74.4	53.7	26.4
sibling	33.6	57.9	39.2	22.1	47.2	82.1	59.9	44.3
avg. Easy P	34.9	64.0	41.0	19.3	39.8	78.1	52.4	35.3
opponent	26.0	26.6	20.0	4.9	8.2	65.9	14.6	16.4
family	41.0	33.5	31.2	8.0	11.3	71.8	19.5	30.6
friend	23.8	26.1	18.7	4.0	11.7	76.1	20.3	21.3
hasMember	25.8	39.6	25.4	10.2	5.9	78.2	11.0	19.5
placeHasPerson	21.3	46.4	23.8	16.7	7.0	83.2	12.9	15.0
avg. Hard P	27.6	34.4	23.8	8.8	8.8	75.0	15.7	20.6
avg. All P	**30.3**	**45.5**	**36.4**	**12.7**	**20.5**	**76.2**	**32.2**	**26.1**

Table 5.17: Drill-Down by Predicate Comparison on *GraphRAG* and *L3X-amp* methods.

on our chosen relations. Especially for hard predicates (which have longer lists with more long-tail entities) like `opponent`, `friend` or `family`, GraphRAG performs poorly (compared to recall and AUC obtained by L3X). Overall, GraphRAG could be an interesting model to compare against, but would likely require integration with other assets (like our re-ranking and batching techniques) for further performance boost.

5.6.8.3 Long-context Models

To assess how well the latest long-context LLMs (Xu et al., 2024b; Li et al., 2025b) can digest book-length texts and cope with the long-list extraction task, we ran 5 different instances of O-list extraction, one for each hard relation and the longest books, feeding OpenAI's latest long-context GPT-4.1 model[8] with the full book texts. The samples were chosen based on the book-length and number of objects in their ground-truth lists.

[8]as of May 2025, https://openai.com/index/gpt-4-1/

Book	Subject	Relation	GPT-4.1		RAG		amp	
			P	R	P	R	P	R
Godfather	Michael Corleone	family (#O=27)	26	33	7	100	14	100
A Song of Ice and Fire	Daenerys Targaryen	Opponent (#O=44)	8	27	2	55	5	86
The Void Trilogy	The Void	placeHasPerson (#O=63)	8	37	14	43	17	43
Harry Potter	Harry Potter	friend (#O=56)	100	78	29	89	25	94
A Song of Ice and Fire	House Baratheon	has_member (#O=142)	70	12	24	15	19	32
Average			42	37	15	60	16	71

Table 5.18: Results (%) comparing L3X with long context models.

The prompt template was formulated as: *"You are a knowledge base. Generate a complete list of names (Objects) who are «predicate definition» of «subject» in the «book» book series. Please list the names one after the other, separated by commas, and only use the given text. «full book»"*. Here, "«»" are placeholders.

Table 5.18 gives comparisons for the precision and recall between the long-context LLM calls and L3X variants (RAG and amp methods). The R@P metrics cannot be computed for the long-context LLM, as it generates a single list without any scores or confidence values.

We observe that long-context based extraction with GPT-4.1 excels at precision but loses big on recall. L3X methods substantially improve recall at the expense of lower precision, and *amp* has consistently higher recall then RAG at similar precision, demonstrating the benefits of the pseudo-relevance feedback strategy.

We further speculate two of the five samples. For the members of *House Baratheon*, GPT-4.1 achieves high precision, but disappointing recall. This suggests that, despite the instruction in the prompt, it does not solely operate on the input text alone, but does also tap into its very rich parametric memory capturing numerous online discussions of the book or TV series (similarly for the friends of *Harry Potter*). For people who live in or travel to The Void (a "place" in the science-fiction trilogy), we observe very low precision and only mediocre recall. As this book has, at best, very sparse coverage in the pre-training contents, this task is fully about understanding the long input text. L3X is far from perfect, but achieves significantly higher recall, with better precision as well.

5.6.9 Error Analysis

We observed recurring error types and discuss three of the most notable cases.

Hallucinations. LLM calls often return huge lists of O's, including names that do not occur in the respective books. Even in RAG mode, the LLM does not necessarily restrict its outputs to entities present in the input passages—a case of *unfaithful* generation. But for recall, our main target, producing names from parametric memory is an advantage. To quantify the effect, we compute the no. of generated O's that do not appear in the respective book:

$$|\cup_{SP}\ \{\text{generated O for SP} \mid O \notin \text{book}\}|$$

normalized by the total no. of generated O values. We observed the following hallucination rates after Stage 1: LLM-only: 55.3%, RAG: 51.7%, amp: 40.7%, amp+neo: 38.1%. Erroneous objects include made-up names and non-entity phrases, such as "X's sister", where the LLM appends a phrase related to the predicate instead of an O name. This underlines the importance of stage 2 scrutinization.

Confusing Predicates. Another common case is that the LLM generates valid O values that are not in the proper relation P with subject S. The most interesting situation here is when that incorrect O is in relation with S for another predicate Q ($\neq$ P) (e.g., Dumbledore appearing among Harry Potter's parents instead of being a friend).

To quantify, we compute a #P×#P confusion matrix, with counts of generated O for P when ground truth is Q. For our best method, *amp/pred(g)*, we observed a ratio of ca. 60:30:10 of accepted true positives (TP), predicate-confused TPs, and accepted false positives (FP). This suggests that merely extracting the right SO pairs is not the problem (only 10% completely FPs), but getting the predicate correct remains a challenge. The most salient predicate pairs of confusion are (`friend`,`family`) and (`friend`,`opponent`). This can be a surprising finding on first glance, but points towards the sophistication and subtlety of fictional literature that makes it difficult for the IE task.

Missing True Positives in the Low Ranks. The majority of TPs are at high ranks, followed by a long tail of mostly FPs but sprinkled with TPs at lower ranks. To assess how well stage 2 recovers *low-ranked* TPs, we use the R@P50 cut-off rank to count the missing TPs below this threshold—i.e., those misclassified as false negatives. Even with our best methods, about 16% of all the ground-truth O values fall into this low-rank, missed-TP category.

5.6.10 Results on the Web Dataset

Table 5.19 shows results for the Web dataset, with different stage-1 configurations, and default *thr80* for scrutinization. By and large, the results reconfirm our key findings with the Books data: LLM-only methods are far inferior; all L3X-RAG methods boost recall; the *amp* has the best AUC after Stage 1 and is the overall winner for R@P after Stage 2. The oracle-based upper bound here is 65% for recall from top-d=500 passages.

A significant difference to the Books data, however, is that all absolute values are substantially lower here (including the oracle). For example, the best R@P50 values (by *amp*) are around 30%, compared to almost 50% for the books experiment. The explanation clearly is that the dataset itself is even more challenging: the ground-truth lists of objects are even longer, with even more long-tail entities, and they are spread across a very large number of web pages—a situation as if all pages (about the same S) were concatenated into a very long, highly incoherent document).

Tables 5.20 and 5.21 compare stage-2 classifiers, along with drill-down by the three predicates. The trends align with those observed in the books dataset, with *amp/pred-g* achieving the highest scores. While performance is strong on the relatively easier `hasCEO` relation (~65% R@P50), it struggles on the highly challenging `hasSubsidiary` relation.

Config	Stage 1			Stage 2 (thr80)				
	P	R	AUC	P	R	AUC	R@50	R@80
LLM-only (zero-ensemble)	25.0	43.5	19.0	28.1	39.7	17.3	25.9	15.5
LLM-only (few-ensemble)	28.3	41.9	18.5	31.8	37.8	16.9	25.2	15.2
RAG	4.1	**70.6**	18.0	4.9	**67.9**	17.8	21.5	7.2
num	4.4	70.3	17.9	5.2	67.9	17.4	20.8	8.3
amp	13.4	60.5	**23.5**	15.7	57.8	**23.3**	**31.9**	**18.3**
amp+neo	18.5	51.8	20.5	**21.0**	48.8	19.5	29.3	13.6

Table 5.19: Results on Web Data. Stage-1 with ranking:{num, amp}; batching:{neo}; k=40; Stage-2: thr80.

Relation	def+thr80			amp+thr80		
	AUC	R@P50	R@P80	AUC	R@P50	R@P80
`hasCEO`	33.5	51.0	19.5	41.1	62.0	42.6
`isMemberOf`	12.1	10.0	1.2	15.9	21.8	8.3
`hasSubsidiary`	7.8	3.4	0.9	12.8	12.0	4.1
macro-avg.	17.8	21.5	7.2	23.3	31.9	18.3

Table 5.20: Results for Web Data with Drill-Down Results by Predicate with *thr80* pruning.

Relation	amp+pred(g)			amp+neo+pred(g)		
	AUC	R@P50	R@P80	AUC	R@P50	R@P80
`hasCEO`	44.7	64.7	44.4	42.9	63.2	42.9
`isMemberOf`	16.4	23.2	9.1	15.6	22.2	9.9
`hasSubsidiary`	13.1	13.0	5.9	12.9	12.3	4.0
macro-avg.	**24.7**	**33.6**	**19.8**	23.8	32.6	18.9

Table 5.21: Results for Web Data with Drill-Down Results by Predicate with *pred* pruning.

5.6.11 Other L3X Configurations

5.6.11.1 Passage Retrieval

We experimented with the following different retrievers in addition to Contriever.

BM25. We used the classic yet effective sparse retrieval model BM25 (Robertson and Zaragoza, 2009) implemented through the Pyserini library[9] (Lin et al., 2021). The queries were formulated with explicit mentions of the subject and relation, complemented by a small set of hand-crafted paraphrases (2–35) and relation-specific cues inspired by query expansion techniques (Voorhees, 1994; Carpineto and Romano, 2012). For example, for `friend` relation, the query was expanded to include paraphrases such as "*companion, supporter, pal, buddy, mate, . . .*" and others.

OpenAI Text Embeddings. Since Contriever imposes a strict 512-token limit on input length, we also evaluated a dense retriever based on embeddings from OpenAI's text-embedding-3-large model[10]. This model accepts inputs up to 8000 tokens and produces embeddings in 3072 dimensions.

In the L3X-RAG configuration, BM25 achieved 83.8% recall, while OpenAI embeddings achieved 84% recall, which is not significantly different from the performance of Contriever 84.3% recall.

5.6.11.2 Passage Reranking

We further experimented with clustering-based passage reranking to promote diversity among retrieved passages. Using the K-Means algorithm (Hartigan, 1979), we clustered similar passages and then applied a round-robin selection by rank across clusters. The rationale here is that diverse passages may contribute to higher recall, whereas highly similar passages can yield redundant cues.

Our clustering procedure involved the following steps:

1. Passage embeddings of dimension d are stacked into matrix $A \in \mathbb{R}^{k \times d}$, and we compute the eigenvalues of $S = AA^T$.

2. On the eigenvalues of S, the Knee Detection algorithm (Satopaa et al., 2011) finds the best number of clusters c.

3. The top-k embeddings are clustered by the K-means algorithm with #clusters = max(5, c).

4. To re-order the top-k passages, we choose b (batch size) passages from each cluster, moving round-robin across clusters.

In terms of Stage 1 performance, clustering-based passage reranking achieved 16.5% precision, 82.0% recall, and 21.1% AUC. As shown in Table 5.7, this method yields higher precision compared to the *num* and *amp* reranking approaches but shows lower recall and the lowest AUC among the three methods.

[9] https://github.com/castorini/pyserini/blob/master/docs/usage-search.md#traditional-lexical-models

[10] https://platform.openai.com/docs/models/text-embedding-3-large

5.7 Demo System

We developed an interactive demo for L3X, designed to support the exploration of multiple configurations. The demo is publicly accessible at: https://d5demos.mpi-inf.mpg.de/l3x. The interface provides an overview of the research work, access to both datasets, and interactive functionality to examine how different L3X configurations—including the baseline systems *LLM-only* and *RAG* termed L3X-def—perform on different inputs such as the choice of book (or series), subject, and relation. In Stage 1, users can inspect generated outputs, drill down into the underlying passages, and perform side-by-side comparisons across configurations. In Stage 2, the *thr* and *pred* classifiers can be run, with the user's choice of threshold or confidence parameters. Evaluation metrics are computed dynamically as users modify selections.

The demo is implemented in Python (Flask) and JavaScript, with list generation powered by the Llama-3.1-70B-Instruct model (Dubey et al., 2024). Outputs are cached to ensure a responsive, interactive experience.

5.7.1 Anecdotal Illustration

As an illustrative example, consider identifying "all opponents of Arya Stark", a prominent character in the five-volume *A Song of Ice and Fire* book series. The top part of Figure 5.4 shows the Stage 1 output using the *amp* configuration. Here, we observe very high recall but at the expense of reduced precision. This demonstrates the necessity of Stage 2, which scrutinizes candidate lists, as depicted in the bottom part of the same figure.

The demo offers drilling down into the internal behavior of L3X. Figure 5.5 illustrates the generated object list for a specific batch of passages. As we see, the *amp* configuration, leveraging pseudo-relevance feedback for retrieving better passages, is able to extract from a highly informative passage featuring Arya Stark's well-known litany of individuals she is in conflict with throughout the book.

5.7.2 Downstream Applications

Extrinsic use cases supported by L3X include corpus-based studies in cultural analytics and digital humanities (Manovich, 2020; Bamman et al., 2024). For instance, L3X can provide the backbone information about character relations and traits, for exploring and analyzing gender issues in literary collections. Inferring gender labels would be straightforward, by spotting personal pronouns and using co-reference resolution within short text spans. This would enable comparing gender distributions across different epochs (e.g., 20th vs. 21st century), cultural contexts, book genres, character roles (e.g., heroes/heroines vs. villains), and more. Also, narrative patterns around certain types of characters can be mined from literature, with support from L3X.

On contents from large Web crawls, a potential downstream application is to automatically construct knowledge graphs for skill allocation, sales, recruiting, and others.

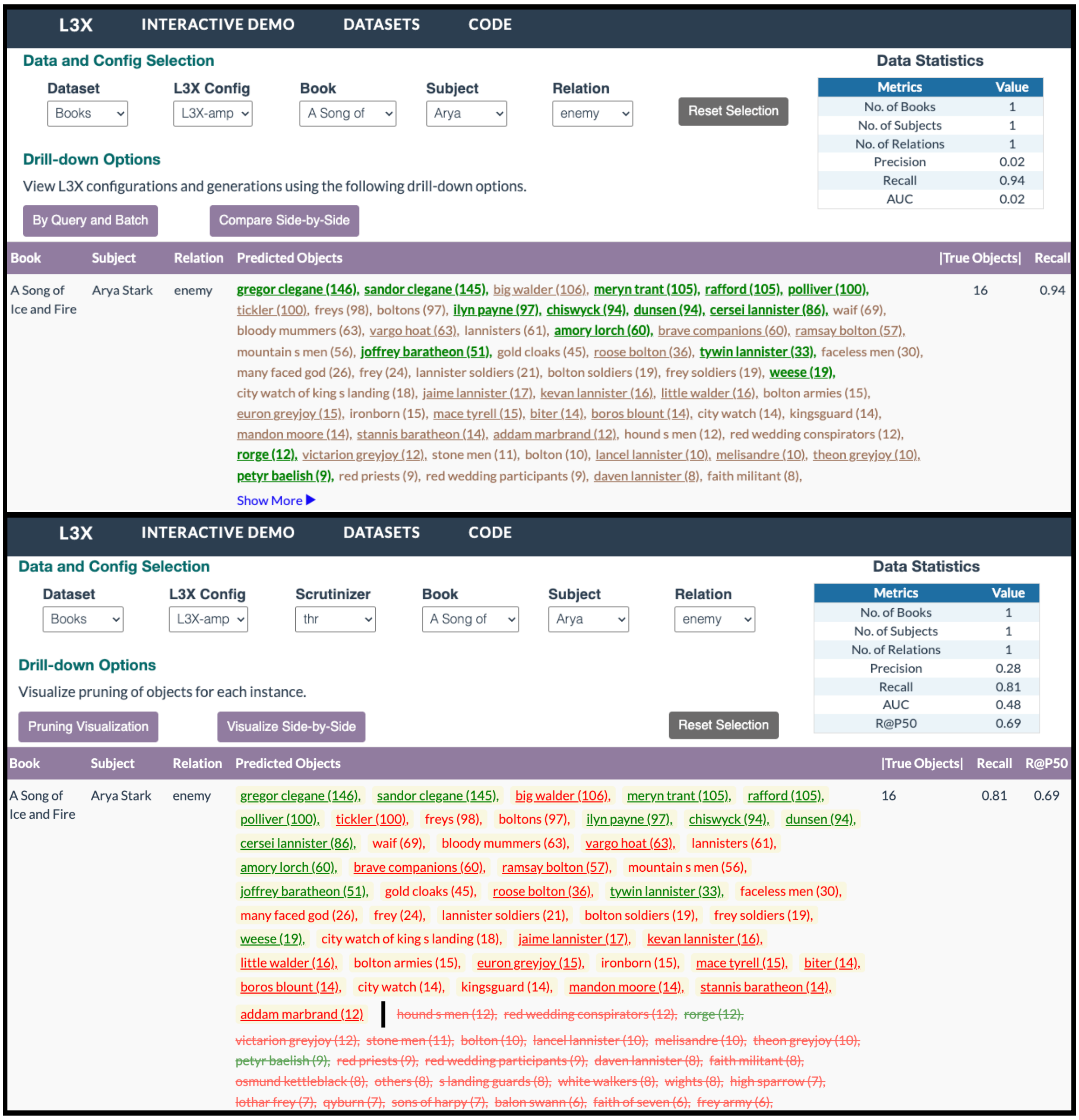

Figure 5.4: L3X-amp results for extracting Arya Stark's enemies: Stage-1 (top), Stage-2 (bottom).

5.8 Related Work

While a detailed background on the relevant topics is given in Chapter 2, this section highlights prior work at the time of this project[11].

Relation Extraction. A common task in IE is to extract the relation P that holds between two given entities, subject (S) and object (O), where P comes from a pre-specified set of possible predicates. State-of-the-art methods (Han et al., 2020; Wang et al., 2020; Huguet Cabot and Navigli, 2021; Xie et al., 2022; Josifoski et al., 2022; Ma et al., 2023b) typically operate on single passages, as input to a multi-label classifier or sequence tagger.

[11] in the year 2024-2025

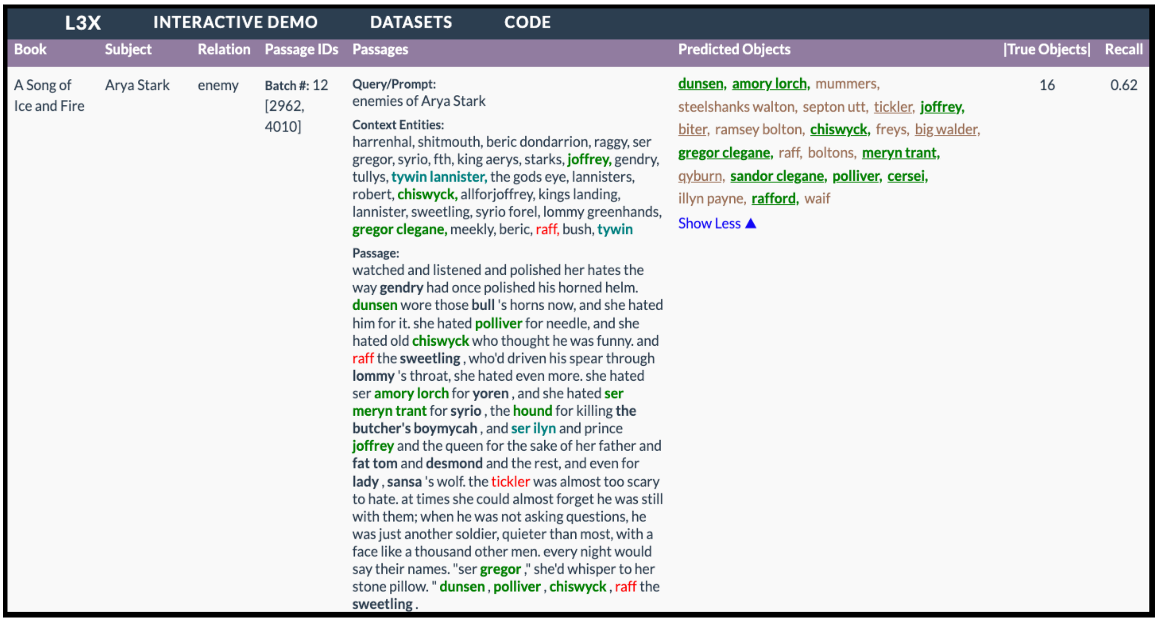

Figure 5.5: Drill-down on passages used for extracting Arya Stark's enemies.

Recent works (Zhao et al., 2024b; Xu et al., 2024a) have advanced the scope of the extractors' inputs under the theme of "long-distance IE", going beyond single sentences/passages. However, techniques like graph neural networks or LLM-powered generative IE are geared for short news or chats, and cannot cope with book-length texts. In the popular document-level benchmark DocRED (Yao et al., 2019), inputs are single paragraphs from Wikipedia. Aggregating cues from many passages (as required, e.g., for determining that Neville is Harry's friend) is out of scope. Moreover, these prior works assume texts for extraction are given upfront, or retrieved by matching S and O in proximity. In contrast, our long-list task takes S and P as input and seeks to generate previously unseen O as output. This changes the goal from high-precision classification to high-recall extraction.

OpenIE. OpenIE (Mausam, 2016; Stanovsky et al., 2018; Kolluru et al., 2022) is a variant where S, P and O are simply surface phrases without linkage to a knowledge base. While OpenIE may provide broader coverage across different relations, it is unsuitable for populating lists of crisp object entities for a given relation. Even when powered by distant supervision with (S,O) pairs, it remains limited to extraction from single sentences or short passages (Smirnova and Cudré-Mauroux, 2019).

LLMs as Knowledge Bases. Petroni et al. (2019) showed that LLM prompts can generate facts of knowledge-base style. The approach has been expanded and refined in various ways (Jiang et al., 2020b; Shin et al., 2020; Qin and Eisner, 2021; Chen et al., 2022b). These aim at precision, disregarding recall and the long tail. Recent studies indicate that LLMs have major problems in dealing with long tail facts (Veseli et al., 2023a; Sun et al., 2023; Singhania et al., 2023b; Kandpal et al., 2023).

Retrieval-Augmented Generation. For better LLM generations, relevant text snippets can be retrieved and fed into in-context prompts, through the popularly known RAG paradigm (Lewis et al., 2020b; Guu et al., 2020). The surveys (Cai et al., 2022; Asai et al., 2023; Wang et al., 2023a; Gao et al., 2023) discuss RAG architectures for improving overall task accuracy.

Evidence and Factuality. LLMs can be harnessed to assess the factuality of statements (Manakul et al., 2023; Min et al., 2023; Chern et al., 2023; Wang et al., 2023a). These techniques leverage external sources, such as Wikipedia articles, which is infeasible in our setting, where the focus is on long-tail entities within long (fictional) books.

5.9 Summary

We introduced the task of extracting long lists of objects from long documents, and proposed the L3X methodology, comprising LLM prompting, retrieval augmentation, passage re-ranking and batching, and classifier-based pruning. Extensive experiments demonstrate that L3X significantly outperforms LLM-only baselines in both recall and R@P. Our best performing L3X configuration *amp-pred(g)*, which leverages pseudo-relevance feedback and a tuned classifier, achieves remarkable performance of ca. 85% recall and ca. 37% R@P80 on full-length books. However, drill-down analyses by predicate and entity popularity reveal substantial gaps in the hard cases. This highlights the core challenge of our task: while scattered textual cues across long books may be intuitive for humans, they remain difficult for AI systems, including LLMs, to reliably detect and extract. Overall, the main takeaways from this research is that recasting information extraction as a list generation task maximizes recall at chosen precision, though 100% precision and 100% recall remains elusive.

6

Information Extraction for Temporal Question Answering

6.1 Introduction

Chapters 3 and 5 examined information extraction (IE) from web documents by defining ground truth as the union of all correct triples and evaluating methods primarily by recall. This chapter focuses on IE for answering temporal user queries, which also falls under the web documents setting, but with an emphasis on the *temporal aspect of information*, and demonstrates how recall-oriented IE enables temporal question answering (QA).

6.1.1 Motivation and Background

Large language models (Peters et al., 2018; Devlin et al., 2019), pretrained on extensive web-scale datasets, have gained widespread usage as human task assistants, offering expert-level knowledge learned during pretraining (OpenAI, 2023a). However, given the dynamic nature of information in the real world, incorporating temporal awareness into these AI systems has become increasingly important. The ability to inject up-to-date information is essential for effectively addressing user needs in a fast-evolving world. Despite the notable capabilities of these systems, research on downstream tasks involving temporal aspects has uncovered several limitations. The key issues include temporal misalignment (Luu et al., 2022; Jang et al., 2022), lack of temporal generalization (Jin et al., 2022), limited temporal reasoning capabilities (Xiong et al., 2024b), and disproportionate handling of implicit versus explicit temporal conditions when answering knowledge-intensive questions (Jia et al., 2021; Kasai et al., 2023).

These challenges are particularly acute in domains such as news, enterprise documentation, and legal or regulatory information, where new—and sometimes conflicting—information emerges from rapidly evolving developments. In such cases, producing reliable, grounded responses cannot rely solely on static, pretrained knowledge. Several studies have demonstrated that LLMs can perform well on domain-specific question answering tasks (Brown et al., 2020; Kaddour et al., 2023), including on popular news datasets. More recent approaches leverage LLMs through the retrieval-augmented generation (RAG) paradigm (Guu et al., 2020; Lewis et al., 2020b; Ram et al., 2023) to address some of the aforementioned limitations. By using trained retrievers to fetch relevant supporting documents, RAG-based methods aim to enhance the accuracy and grounding of generated responses.

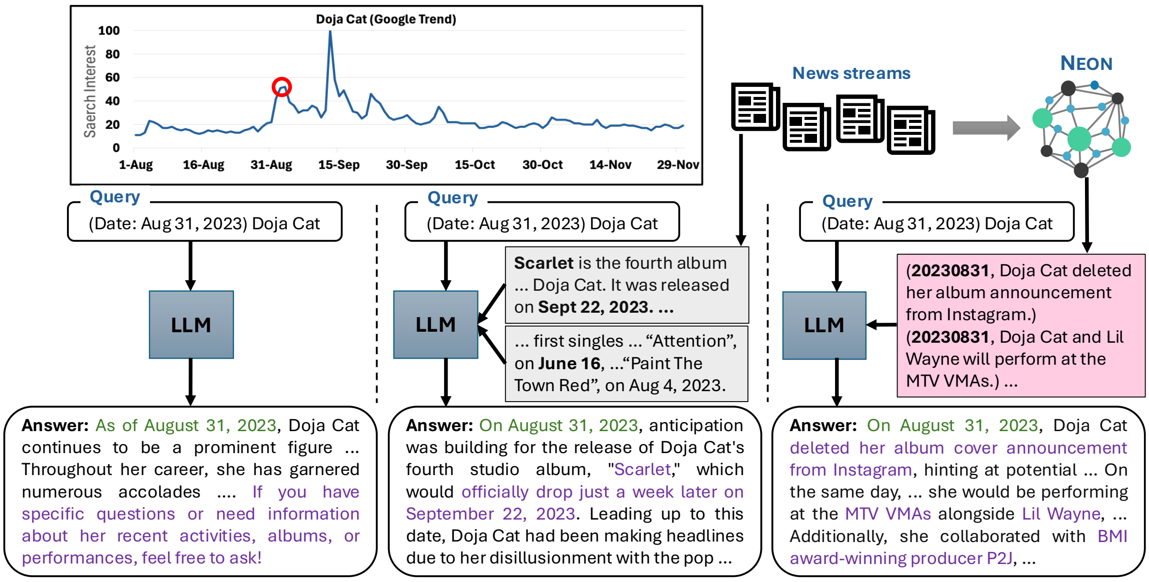

Figure 6.1: Example of entity-centric, time-specific QA. The graph shows search interest for *Doja Cat* over a four-month period. The bottom part illustrates response generation at one of the peaks (*31 August*) using three different techniques: (i) zero-shot prompting, (ii) news snippets based prompting, (iii) augmenting tuples from our NEON graph for enhanced answer generation.

The aim is to generate reliable, temporally grounded propositions to support user queries, with a particular emphasis on *entity-centric queries*. This choice is motivated by several factors:

1. Users frequently inquire about entities during real-world events (Guo et al., 2009; Lin et al., 2012), often with explicit or implicit time constraints.

2. User queries are often vague, ambiguous, and telegraphic, with intent implicitly tied to the mentioned entities (Sawant and Chakrabarti, 2013; Balog, 2018).

3. Entity-centric information tuples of the form (`subject`, `predicate`, `object`) are highly valuable in knowledge-base-powered question answering tasks (Jia et al., 2021; Mavromatis et al., 2022) and serve as compact, indexable units of evidence that bridge unstructured news text and structured reasoning.

For instance, Figure 6.1 shows the Google trend line[1] for the popular entity *Doja Cat* during the *August-November, 2023* timeframe, exhibiting peak search interest on specific dates, particularly around major events, e.g., on *August 31* and *September 15*. When a query such as "Doja Cat" or "Doja Cat news" is issued as a prompt to an LLM such as GPT-4 (OpenAI, 2023a), the model generates an entity-centric response based on sources like Wikipedia, one of the largest resources in LLM pretraining datasets (Raffel et al., 2020; Weber et al., 2024). Although Wikipedia provides valuable overviews and highlights about key events, it often lacks timely updates required to answer entity-centric queries involving recent developments. Even if the user query is reformulated to specify

[1] `https://trends.google.com/trends/explore?cat=3&date=2023-08-01%202023-11-30&q=Doja%20Cat`

a date—e.g., "(Date: August 31, 2023) Doja Cat"—LLMs may not consistently produce precise, up-to-date responses, especially when the date is beyond the model's knowledge cutoff and the system must defer to external sources through RAG mitigation.

This limitation is illustrated in Figure 6.1, where we select a peak in the search interest plot. Without additional information sources and/or additional context, GPT-4 provides a generic response to the prompt "(Date: August 31, 2023) Doja Cat". However, we observe improvements in temporal relevance and accuracy through RAG mitigation when supporting snippets retrieved from news streams are incorporated into the LLM's prompt. Here, the LLM still needs to reason and extract relevant information across these multiple snippets.

6.1.2 Research Questions

The goal of this work is to extract OpenIE-style temporal propositions from news streams and use them for answering use queries. This entails answering the following research questions:

RQ1: How can we process and utilize information from large-scale news streams? (Section 6.2.1.1)

RQ2: How well do LLMs operating in RAG mode, perform on this temporal answer generation task? (Section 6.4)

RQ3: How do we effectively measure the quality of responses for entity-centric queries in the absence of ground truth? (Section 6.3.2)

6.1.3 Approach

As Liu et al. (2024) note, both the size of the context and the positioning of relevant information within the context significantly impact model performance. Instead of directly augmenting the prompt with chunks of information, we propose extracting entity-centric propositions, which would capture interactions such as events and activities between entities in a more compact form (single sentences or propositions), and uses these for enhanced response generation.

Traditional IE pipelines, including those powered by LLMs, typically perform heavy-duty processing on input documents, involving entity recognition, relation extraction, and post-extraction de-duplication. These steps are necessary because such systems prioritize precision, often at the cost of recall (Han et al., 2020). In contrast, our methodology, termed NEON (**N**ews **E**ntity Interacti**ON**s), begins by pre-processing the news stream to identify, disambiguate, and label entities directly in the text. We then incorporate entity markup into LLM prompts to generate Open Information Extraction (OpenIE)-style entity interactions, achieving higher recall (Manning et al., 2014; Kolluru et al., 2020a). The extracted propositions are then used for answering temporal user queries. For instance, in Figure 6.1, two propositions related to *Doja Cat* are augmented for response generation.

We evaluate the effectiveness of NEON over 3,000 real-world queries, randomly sampled from Bing search engine logs. Each query was required to explicitly mention one of 50 sampled entities. These entities are chosen with various considerations in mind, including demographic diversity, the balance between popular and long-tail entities, and a mix of person- and organization-type entities, as explained further in Section 6.3. To perform temporal QA generation, we augment the

LLM prompts with retrieved propositions. Given prior work on using LLMs as evaluators (Liu et al., 2023b; Wang et al., 2024), particularly on downstream tasks within the news domain (Xu et al., 2023; Chiang and Lee, 2023), we use *LLM-as-a-Judge* to automatically assess the quality of responses on a 3-point Likert scale. We also corroborate the automatic evaluations through human assessment. Our experiments reveal that integrating propositions into LLM prompts improves the temporal relevance and overall quality of the answers.

6.1.4 Contributions

The salient contributions of this work are:

1. We propose Neon, a methodology for extracting OpenIE-style propositions from news streams. Utilizing these propositions rather than raw document chunks leads to enhanced temporal question answering.

2. We curate a dataset of 3,000 real-world temporal queries and demonstrate that proposition-based augmentation outperforms standard retrieval baselines in accuracy and grounding.

3. We provide granular evaluation across three aspects: *helpfulness, relevance, faithfulness*, offering insights into how human assessments rely on commonsense reasoning and context, whereas LLM-based automatic evaluation strictly adheres to prompt instructions.

6.2 Methodology

We present Neon, which extracts OpenIE-style propositions to model entity interactions from news streams. Table 6.1 provides illustrative examples from two configurations. We outline the extraction pipeline in Section 6.2.1. Section 6.2.2 describes the repository of propositions for efficient temporal retrieval. Finally, Section 6.2.3 details our approach for temporal question answering, demonstrating how Neon enables contextually relevant and grounded responses.

6.2.1 Entity-Interactions Extraction

6.2.1.1 Data Pre-Processing

Neon propositions are assertions or statements in the style of a knowledge graph, where nodes are entities and edges capture interactions (events or activities) between these entities. To extract these propositions, we process a large collection of news streams from various internet sources. Each article is timestamped according to its publication date, extracted from the source URL. The data pre-processing comprises the following steps:

1. **Entity identification:** Each news article is preprocessed to extract only the main content, which is then analyzed using an entity linking tool called NEMO (Cucerzan, 2014). This step annotates all named entity mentions in text, including co-references and additional information like dates and addresses, with a simple XML markup.

Method	(Timestamp, Proposition)
NEON-S	(*20230502*, **Doja Cat** made debut appearance at **Met Gala**) (*20230502*, **Doja Cat** dressed as **Choupette**) (*20230502*, **Doja Cat** was styled by **Brett Alan Nelson**) (*20230831*, **Doja Cat** will perform at the 2023 **MTV VMAs**) (*20230831*, **Doja Cat's** announcement debuting the cover art for "Scarlet" was removed from **Instagram**)
NEON-SP	(*20230502*, **Doja Cat** and **Jared Leto** paid homage to Lagerfeld) (*20230531*, **Doja Cat** and **Demi Lovato** will perform at VMAs) (*20230831*, **Doja Cat** collaborates with Afrobeats producer **P2J**) (*20230831*, **Doja Cat** posted on **Twitter**) (*20230831*, **Doja Cat** deleted a post on **Twitter**)

Table 6.1: Samples from NEON variants. The subject and object entities are bold-faced.

2. **Sentence segmentation:** The content is segmented into sentences using a proprietary tool[2], similar to spaCy[3], and entities identified in previous step are tracked within each sentence.

3. **Text chunking:** The contain is then split into overlapping chunks. These chunks are indexed along with related metadata, including the set of named entities identified within each chunk and the article's timestamp.

4. **Chunk deduplication:** To manage paraphrased information across different chunks, near-duplicate chunks are filtered out using trigram representations and Jaccard similarity on entity sets (with a threshold $t = 0.85$, a tuned hyperparameter).

Once the set of news chunks is obtained, LLM-based information extraction (IE) techniques are applied to each chunk for proposition extraction.

6.2.1.2 Proposition Extraction

We design task-specific LLM prompts to explore two IE variants for proposition extraction. In the first variant, termed NEON-S (see Section 6.2.1.3), the LLM generates propositions using only the subject entity s and its corresponding text chunk. In the second variant, termed NEON-SP (see Section 6.2.1.4), extraction is conditioned on both a subject s and an object o, targeting chunks where the pair (s, o) co-occurs.

NEON-S focuses on interactions centered on a single subject entity, making it suitable for queries where the entity (e.g., a person or organization) is the primary focus. Conversely, NEON-SP enables the model to infer direct relationships or events between an (s, o) pair. Both approaches perform open extraction without relying on predefined relation schemas. Given that temporal QA is our

[2]This work was carried out during an internship at Microsoft Research and an internal tool was utilized.
[3]`https://spacy.io/api/sentencizer`

User: Given the news snippet, extract and summarize the interactions of {subject} with other named entity (names within <e> and </e> tags) mentions. Use as few words as possible. Per line generate a single interaction between {subject} and the other named entity.

Snippet:
{snippet} May 2, 2023: Stepping out at the <e> Met Gala </e> 2023 (her first ever!), <e> Doja Cat </e> instantly delivered one of the evening's most Internet-breaking moments by channeling the one and only <e> Choupette </e>, <e> Karl Lagerfeld's </e> beloved cat. "It makes no sense for anyone to go as <e> Karl's </e> cat more than <e> Doja Cat </e>," says <e> Brett Alan Nelson </e>, her creative director and stylist ...

Figure 6.2: Prompt Template for Neon-S.

downstream evaluation task, we instruct the LLM to generate natural language propositions (Oguz et al., 2022; Jia et al., 2024), thereby enabling direct prompt augmentation for downstream usage.

6.2.1.3 Subject-centric Neon

The primary objective of the Neon-S variant is to automatically construct an entity-centric, timestamped KG using a target set of subjects provided by the user (details in Section 6.3), facilitated through LLM prompting. Given a news chunk, the model identifies neighboring nodes (object entities already marked in the text) and the connecting edges (interactions or propositions) between the subject and object entities. The process consists of the following steps:

1. **Target Subjects:** Begin with the target set of subjects.
2. **Chunk Retrieval:** For each subject, retrieve all the news chunks that mention the subject.
3. **Prompt Construction:** For each subject, construct a prompt by explicitly highlighting the subject; inserting the corresponding news chunk along with its timestamp; and marking all entities in the chunk using "<e>" and "</e>" tags.
4. **Proposition Extraction:** Use the prompt to extract a list of natural-language propositions, and assign each proposition the timestamp of its corresponding chunk.
5. **Iteration:** Repeat the above steps for all subject-chunk pairs to iteratively build the KG.

Although this approach emphasizes information related to the target subject entities, scaling up the set of subjects can lead to a more diverse and complete KG. The chunk size can be large enough to provide detailed context around the subject while still fitting within the LLM's context window. This setup allows a single prompt to generate a list of propositions. The prompt template for Neon-S is shown in Figure 6.2.

6.2.1.4 Subject-Object-Pair-Centric Neon

In contrast to Neon-S, the idea behind Neon-SP is to leverage the co-occurrence of entity mentions within news chunks as potential subject–object pairs. We start with a predefined set of subject

User: Given the news snippets, extract and summarize the interactions between {subject} and {object}. Use as few words as possible. Per line generate a single interaction between {subject} and {object}.

Snippets:
[1] {snippet} May 1, 2023: <e> Doja Cat </e> and <e> Jared Leto </e> deliver <e> Lagerfeld </e> -inspired looks. For this year's theme, the A-list guests were asked to wear outfits which paid homage to the German fashion designer
...

[k] {snippet} May 2 2023: ... To be fair, <e> Leto </e> was not the sole <e> Choupette </e> on the carpet. <e> Doja Cat </e> went the Animorphs route, wearing ... cap with her <e> Oscar de la Renta </e> gown. It's almost like <e> Doja Cat </e> and <e> Jared Leto </e> went to the Spirit Halloween store together ...

Figure 6.3: Prompt Template for Neon-SP.

entities, identify entity co-occurrences in the news chunks, and then determine candidate pairs based on their TF–IDF scores. Once the target pairs are obtained, the LLM is prompted to perform proposition extraction. The overall process mirrors that of Neon-S, but with different prompts. Specifically, the steps are as follows:

1. **Target pair selection:** Begin with a target set of subjects. For each subject, identify co-occurring entity pairs within all news chunks whose TF–IDF scores are high (≥ 0.8). These pairs become the targets for interaction extraction.

2. **Chunk retrieval:** For each subject–object pair, retrieve all relevant news chunks and their associated timestamps. Sort the retrieved chunks by timestamp and process them in batches of size k (a hyperparameter), such that each batch fits within the LLM's context window.

3. **Prompt Construction**: Construct a prompt by explicitly highlighting the subject-object pair; inserting the corresponding batch of news chunks along with its timestamps; and marking all entities in the chunks using "<e>" and "</e>" tags.

4. **Proposition Extraction:** Use the prompt to extract a list of natural-language propositions, and assign each proposition the timestamp of its corresponding chunk.

5. **Iteration**: Repeat the steps for all subject-object-chunk batches to iteratively build the KG.

This variant often yields more generations due to the explicit subject-object pairs. However, it can cause the LLM to generate propositions involving (s, o) without a valid interaction, since the entities co-occur and the prompt explicitly asks the model to generate, leading to hallucinations when no such link exists in the text. It can also lead to an empty list extraction if no interaction is detected. Figure 6.3 shows the prompt template for Neon-SP. Notably, the chunk batch size k directly affects performance: a small k is leads to more LLM calls and reduced contextual detail, while a large k risks "lost-in-the-middle" (Liu et al., 2024) or information overload as observed in the previous chapter. Table 6.1 provides illustrative propositions from both variants for the subject *Doja Cat*.

6.2.2 Temporal Repository

We represent the NEON propositions in a datastore where each entry is a tuple (t_d, d): d is a natural-language proposition and t_d is its associated timestamp. Following prior work (Li and Croft, 2003; Berberich et al., 2010; Neelakantan et al., 2022; Izacard et al., 2022), we employ off-the-shelf embedding models, such as OpenAI's `text-embedding-3-large`[4], to create dense indexes for efficient retrieval. User queries in our dataset (details in Section 6.3.1) are encoded into dense vectors using the same model, allowing retrieval of the top-k relevant propositions via cosine similarity. This approach is retriever-agnostic and can be integrated with any suitable retrieval model.

To ensure temporal relevance when retrieving content for answering user queries, we reformulate each query to explicitly specify its timestamp. We then experiment with two retrieval strategies:

1. **Temporal Retrieval**: Retrieve the top-k propositions using the original user query, where the proposition timestamp exactly matches the query timestamp. If fewer than k propositions match, retrieve additional tuples within a $\pm r$ day window around the query timestamp to complete the top-k results, where r is a hyperparameter.

2. **Generic Retrieval**: Retrieve the top-k propositions solely based on semantic similarity between the query and the propositions in the datastore.

Temporal retrieval can improve relevance for time-specific queries but may yield zero propositions when news coverage is sparse for certain dates. In contrast, generic retrieval is more robust: it first attempts to retrieve temporally aligned propositions and, if necessary, backs off to semantically relevant results (Oard et al., 2001; Resnik et al., 2001).

6.2.3 Temporal Question Answering

To demonstrate the utility of the extracted propositions, we evaluate their effectiveness on the downstream task of temporal question answering. The pipeline proceeds as follows:

1. The user issues a (telegraphic) query at a specific timestamp.

2. The query is reformulated by performing named-entity disambiguation (Cucerzan, 2007) and by explicitly mentioning the timestamp in natural language.

3. Using the reformulated query, we apply the retrieval methods described in Section 6.2.2 to obtain the top-k relevant propositions.

4. The top-k propositions are augmented into an LLM prompt for retrieval-augmented generation.

[4] https://platform.openai.com/docs/models/text-embedding-3-large

5. The LLM generates a response to the temporal query using the retrieved propositions as supporting evidence.

Figure 6.4 provides an overview of this temporal QA pipeline. The goal is to generate responses that are both temporally relevant and well grounded in the underlying propositions.

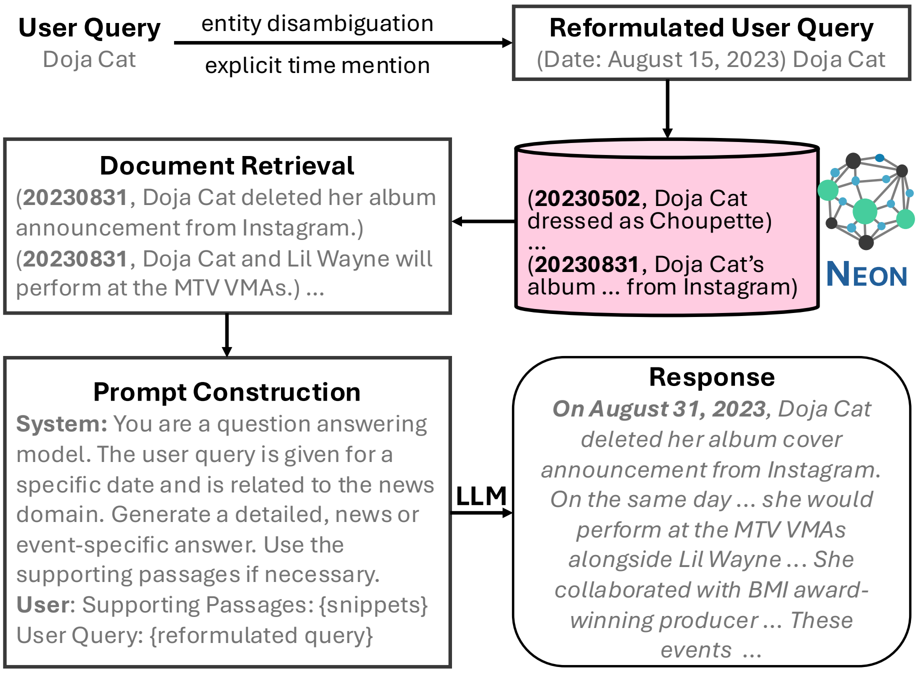

Figure 6.4: Temporal Question Answering Pipeline

6.3 Experimental Setup

6.3.1 Dataset

Entities. We selected 50 target subject entities via stratified sampling across four categories—artists, companies, leaders, and pioneers—from Time's 100 most influential people[5] and Fortune500 companies[6] to ensure a balanced diversity in category, demographics, and popularity.

User Queries. For each entity, we collected a large volume of user queries from Bing search logs over a four-month period (`August-November, 2023`). Because we aim to generate temporal answers for emerging events, we focus on entity-centric queries logged during *spiking dates*—periods of heightened entity presence in user queries. Spiking dates are identified through short-term trends in query volume. Specifically, we compute a 3-day rolling sum of daily query counts and flag significant spikes as dates where the rolling sum exceeds one standard deviation above the mean. This threshold effectively captures periods of elevated user interest, often triggered by shifts in attention or external events related to the entity. To preserve user privacy, the queries issued by

[5]`https://time.com/collection/100-most-influential-people-2023`
[6]`https://fortune.com/ranking/fortune500/2023/`

fewer than five distinct individuals are filtered out, yielding nearly 3000 temporal user queries. Table 6.2 provides representative examples from each category and popularity segment.

	Entity	Type	Date	Query
Artists	Neil Gaiman	tail	Sept 5	neil gaiman award
			Sept 5	which novel did neil gaiman coauthor with terry pratchett
	Austin Butler	head	Aug 17	austin butler as elvis
			Aug 18	what are the latest projects of austin butler
Companies	Berkshire Hathaway	tail	Nov 6	berkshire hathaway cash pile
			Nov 7	berkshire hathaway stock
	Alphabet Inc.	head	Aug 24	alphabet stock price today
			Sept 14	alphabet layoffs
Leaders	Sherry Rahman	tail	Aug 25	climate financing by rahman
			Oct 5	pak minister sherry rehman
	Olaf Scholz	head	Sept 3	olaf scholz falls
			Sept 4	olaf scholz eye patch
Pioneers	Bella Hadid	tail	Oct 27	bella hadid cancelled
			Oct 31	bella hadid closing coperni
	Doja Cat	head	Aug 28	doja cat beefs with fans
			Aug 29	doja cat feels free

Table 6.2: Samples of User Queries

News Articles. To extract propositions, we collect news streams from 500 distinct sources, with a median of 200 articles per source and a maximum of 9,000 articles. Articles vary in length and are preprocessed to construct metadata, including disambiguated named entity mentions (e.g., persons, organizations, and locations), their surface forms, and article timestamps. This metadata enables mapping each target subject to its corresponding news articles. We also identify duplicate content across articles and publication dates, and retain only unique content mapped to all associated sources and dates. Finally, to ensure robustness across varying levels of media coverage, we stratify entities into head (high-frequency) and tail (low-frequency) groups based on the 70th percentile of total article counts. Table 6.3 summarizes the final dataset statistics.

Figure 6.5 shows the number of unique news sources per subject entity, grouped by the four categories. Dashed lines at 25%, 50%, and 75% on the y-axis mark the quartiles, providing a visual reference for comparing news coverage across entities. Distinct patterns emerge by category. Leaders receive the highest overall coverage: e.g., "*Joe Biden*" and "*Mitch McConnell*" appear in all 500 sources, reflecting their strong news presence. Companies follow closely, with entities like "*UnitedHealth Group*" and "*Berkshire Hathaway*" covered by 300-400 sources, and some exceeding

Entity sources	Time100 and Fortune500
No. of subjects	50
No. of head subjects	15
No. of tail subjects	35
No. of news domains	500
Median no. of news	~200
Max no. of news	~9000
No. of user queries	3000
Spiking date range	August–November 2023

Table 6.3: Dataset Statistics

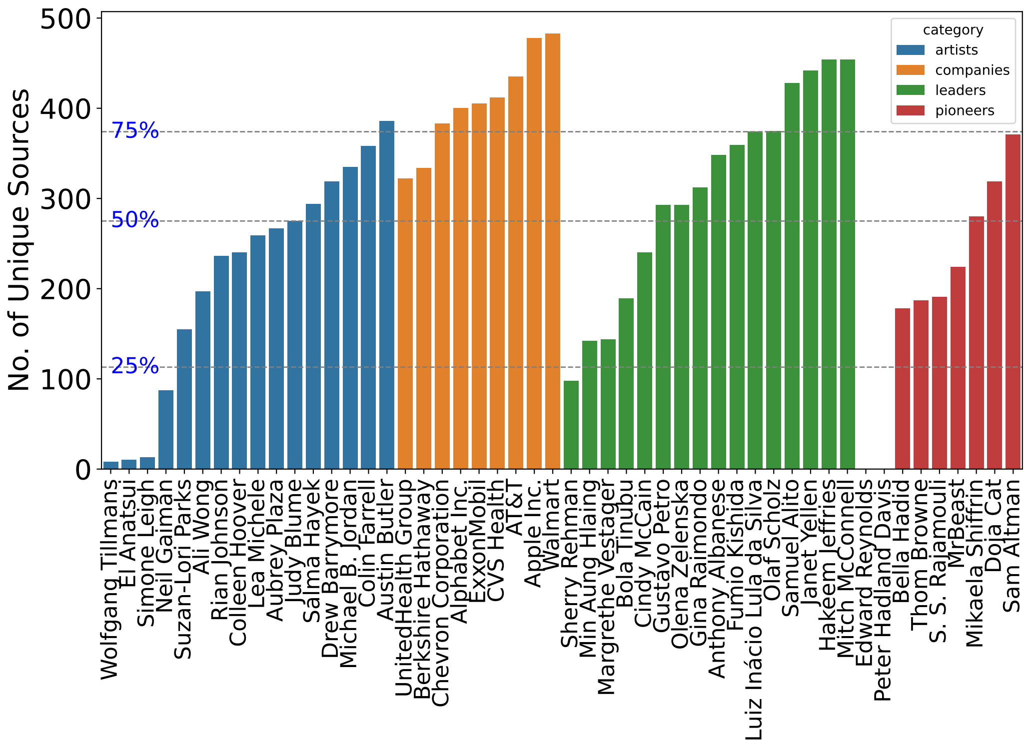

Figure 6.5: Coverage for 50 entities in 500 diverse news sources over a period of one year (2023)

400, indicating substantial coverage. Artists generally appear in 100-300 sources, suggesting that while culturally impactful, their coverage is less pervasive. Finally, Pioneers exhibit the most varied coverage, ranging from about 50 to just under 400 sources, with figures such as "*Sam Altman*" and "*Doja Cat*" drawing attention around specific events.

6.3.2 Evaluation Setup

Since the temporal QA task involves generating textual responses for evolving news events, traditional evaluation methods are challenging due to the absence of an established ground-truth dataset. To address this, we employ an automated evaluation framework that leverages LLMs, which have recently gained prominence (Kocmi and Federmann, 2023; Wan et al., 2024), even in news-related tasks (Xu et al., 2023; Chiang and Lee, 2023), as an efficient means for output assessment. Our evaluation focuses on three key aspects:

1. *Helpfulness*: assessing the extent to which the response addresses the user's query by incorporating additional relevant entities—such as people, events and locations—beyond the primary subject

2. *Relevance*: ensuring that the response is accurately aligned with the context of the query, delivering information that is specific to the date or timeframe referenced

3. *Faithfulness*: evaluating the extent to which the response is grounded in the supporting passages provided, relying exclusively on verifiable information from these sources to produce a trustworthy answer.

Each aspect is rated on a 3-point Likert scale, in alignment with established practices in prior evaluation studies (Wang et al., 2024; Maddela et al., 2021). Notably, work by Dettmers et al. (2023) observed that the order and detail of aspect ratings can influence evaluation outcomes. To mitigate position bias, we prompt the LLM (GPT-4o[7]) separately for each aspect, tripling the number of LLM calls but reducing the risk of cross-aspect contamination. To ensure robustness in our evaluation, we perform few-shot evaluation using human annotated in-context examples on the same aspects. The prompt templates for evaluation are provided in the Appendix A.3.

6.3.3 Baselines and Neon Configurations

We augment LLMs with temporally relevant information for answering entity-centric questions. The following methods are evaluated within the RAG framework:

1. **NewsRAG:** Retrieves the top-k temporally relevant news chunks, which serve as the source for extracting propositions in the Neon methodology (step 2 in each variant of Section 6.2.1).

2. **WebRAG:** Uses Bing to retrieve top-k web snippets, explicitly incorporating timestamps to access diverse and dynamic sources.

3. **Neon-S:** Employs the process in Section 6.2.1.3 and includes 109,246 proposition entries.

[7] https://openai.com/index/hello-gpt-4o/

	Method	Helpful	Relevant	Faithful
Temporal	NewsRAG	1.03	0.83	0.83
	Neon-S	1.03	0.82	0.86
	Neon-SP	1.04	0.79	0.87
Generic	NewsRAG	0.9	**1.24**	1.08
	Neon-S	**1.38**	1.12	**1.18**
	Neon-SP	1.37	1.03	1.17
	WebRAG	1.54	1.29	1.22

Table 6.4: Performance with zero-shot evaluation prompts.

	Method	Helpful	Relevant	Faithful
Temporal	NewsRAG	1.53	1.45	1.46
	Neon-S	1.53	1.50	**1.52**
	Neon-SP	1.55	**1.54**	1.56
Generic	NewsRAG	1.59	1.41	1.05
	Neon-S	1.42	1.34	1.05
	Neon-SP	1.45	1.35	1.09
	WebRAG	**1.57**	1.57	1.41

Table 6.5: Performance using few-shot learning and reasoning based evaluation prompts.

4. **Neon-SP:** Employs the process in Section 6.2.1.4; this variant includes 455,680 proposition entries.

Each method uses the reformulated user query for retrieval, augments the top-k retrieved information directly into the prompt, and employs GPT-4o[8] for response generation. Other OpenIE-based methods are unsuitable for our purposes as they explicitly model knowledge tuples, which we aim to avoid. Comparing the performance across these methods reveals how different forms of real-time information impact helpfulness, relevance, and faithfulness in addressing the temporal demands of our QA task.

6.4 Results

Tables 6.4 and 6.5 presents the main results, comparing system performance across *temporal* and *generic* retrieval strategies using a 3-point Likert scale across the evaluation aspects. WebRAG is invariant to retrieval strategy since Bing combines features of both temporal and generic retrieval. Best scores are highlighted, while second-best scores are boldfaced.

In Table 6.4, we notice that in zero-shot setting WebRAG achieves the highest performance across all three aspects; this is expected considering that it is achieved with the help of a search

[8] https://openai.com/index/hello-gpt-4o/

engine. Meanwhile, NEON variants consistently outperform NewsRAG on both retrieval strategies for the helpfulness and faithfulness aspects, demonstrating that temporally relevant NEON KB entries improve generation reliability. Between NEON variants, both perform similarly on helpfulness and faithfulness, but NEON-S achieves a higher relevance scores.

Few-shot learning with in-context examples and chain-of-thought prompts have been effective for downstream tasks using LLMs (Wei et al., 2022). We hand-craft an in-context example for each evaluation aspect and Likert scale combination to demonstrate to the LLM what a likely score should be, given the user query, the temporally grounded answer generation, and the aspect-specific evaluation criteria. These examples are constructed using queries for subject entities that do not appear in the test set. In Table 6.5, unlike zero-shot evaluation, few-shot results show that temporal retrieval outperforms generic retrieval across all methods. Additionally, higher gains are observed with NEON and NewsRAG methods, and the results compare to those of WebRAG in many instances.

Each method shows specific strengths: NewsRAG produces the most helpful generations, WebRAG excels at relevance due to its broad and diverse sources, while NEON delivers the most reliable outputs. NEON-SP specifically benefits from a larger KB (4x the size of NEON-S) and captures diverse entity-interactions by explicitly using subject and object mentions employed in KG construction. Moreover, automatically generating relevant few-shot examples for such an LLM-judge task remains an open area for future work.

6.4.1 Human Assessment

To better understand and validate the automated evaluation results, we conducted a human assessment focusing on our 3 evaluation aspects. Annotators[9] rated 100 randomly sampled responses across all methods using a 3-point Likert scale. To ensure consistency, annotators were provided with detailed descriptions of each aspect, method-specific supporting documents included in the LLM prompt for temporal QA, and examples illustrating each aspect-scale combination. External searches were not allowed, and ratings were based solely on the reformulated query, supporting documents, and LLM-generated responses.

The Pearson correlation between human assessments and few-shot prompt-based automated evaluations showed weak positive correlations: 0.24 (helpfulness), 0.12 (relevance), and 0.19 (faithfulness). Inter-annotator reliability, measured on the overlapping 10% of the responses from 3 annotators using Krippendorff's alpha (Krippendorff, 2011), indicated strong agreement with scores of 0.71 (helpfulness), 0.77 (relevance), and 0.61 (faithfulness).

A drill-down analysis of the ratings revealed that discrepancies between automated and human scores often stemmed from mismatches in timestamps across the query, supporting passages, and LLM responses. Automated evaluations consistently assigned a score of 0 in these cases, whereas human annotators used common-sense reasoning and contextual understanding of the dates to assign scores of 1 or 2.

Interestingly, the ratings for helpfulness and relevance, both human and automated, exhibited a high correlation (>0.8). This is likely because a highly relevant response—one that accurately

[9] The author of this thesis and two other employees at Microsoft

reflects the query's specific timeframe—almost always leads to a helpful response. The reverse however is not always true, as a helpful response may not strictly adhere to temporal relevance.

6.5 Analysis and Discussion

We perform a qualitative analysis to gain insights into the successes and failures of each method. We identified examples that either support or hinder the generation of temporally relevant responses. Tables 6.6 and 6.7 provide illustrative examples.

Entity-centric temporal RAG enhances reliability. We leverage an entity-centric temporal datastore, which improves retrieval accuracy. This is evident in methods using news chunks (NEON variants and NewsRAG), as Bing search in WebRAG relies on surface form matches that often retrieve irrelevant snippets, particularly for long-tail entities with low search hit rates (example 1 in Table 6.6). Consequently, the entity-centric design of NEON leads to more reliable responses.

WebRAG benefits from non-English information and LLMs. We notice that Bing search sometimes retrieves noisy passages together with relevant passages that are in languages other than English, particularly for subject entities from non-English demographic regions. In such cases, the LLM is still able to generate temporally relevant responses in English based on the non-English passages (example 2 in Table 6.6). This gives WebRAG an advantage over others, as the news streams we processed are in English.

Deviations from rating instructions. We observe that LLMs do not always adhere to the evaluation instructions. Despite the prompt clearly outlining the criteria for a 3-point Likert rating, the model generates ratings outside the intended range along with reasons for such ratings. A response is assigned a rating of 3 when the LLM reasons it as somewhat relevant, and other responses with ratings of 5 and 9 when they are deemed reliable and relevant (shown in Table 6.7).

6.6 Related Work

While a detailed background on the relevant topics is given in Chapter 2, this section highlights prior work at the time of this project[10].

Information Extraction. Extracting structured information from unstructured text has been a central focus of NLP research (Han et al., 2020). A key task in IE is relation extraction (RE), which identifies relationships between entities using named entity recognition, entity linking, and relational classification. Accurate relation identification, however, requires deep semantic understanding of context and is often constrained by the fixed relation types in traditional classifier-based RE methods (Ma et al., 2023b).

To address these limitations, OpenIE enables entities and relations to be extracted as surface forms or phrases (Kolluru et al., 2020a). Both RE and OpenIE are vital for constructing and enriching knowledge graphs (Martínez-Rodríguez et al., 2018; Singhania et al., 2022b). Recent advances explore LLMs as implicit KBs (Petroni et al., 2019), leveraging their outputs for IE tasks via prompting (AlKhamissi et al., 2022; Hao et al., 2023; Cohen et al., 2023). Our work leverages LLMs

[10] in the year 2024

	Query	Date	Method	Retrieved Passages	Temporal QA Response
example 1	Sherry Rehman news	18 Oct	WebRAG	[1] Waheeda Rehman Biography ... born in Chengalpet ... [2] Sherry Rehman, parlementaire du Pakistan ... [3] For a sweeter white ... substituting sherry for white wine ... [4] PX Sherry Cask Matured limited whisky ... [5] Waheeda Rehman is an Indian ...	On October 18, 2023, Sherry Rehman, a prominent member of the Pakistan People's Party, is in the news due to her involvement in a legal case concern ... This development highlights the ongoing and contentious debate ... those who seek to reform them.
example 2	Olaf Scholz eye patch	4 Sept	WebRAG	[1] Chancellor 'Pirate Olaf' Scholz injured jogging ... [2] ... Olaf Scholz. Can someone Photoshop an eye patch ... [3] ... lors d'un footing ce samedi 2 septembre 2023, le chancelier allemand Olaf Scholz s'est ... [4] Olaf Scholz Pirat is not real ... [5] Olaf ... ready for the great pirate era ...	On September 2, 2023, German Chancellor Olaf Scholz sustained minor facial injuries while jogging ... he appeared in public on Monday, September 4, 2023, wearing an eye patch, which has led to a wave of ... playful nickname "Pirate Olaf" among social media users ...

Table 6.6: Examples of queries and LLM responses using retrieved passages

	Query	Date	Method	Temporal QA Response	LLM Rating
example 3	Sam Altman Microsoft	22 Nov	NEON-SP	As of November 22, 2023, Sam Altman has been significantly involved with Microsoft ... OpenAI, which Altman co-founded, is 49% owned by Microsoft ... In summary, as of late November 2023, Sam Altman continues to play a pivotal role ..	{"reason": "The response provides a detailed and relevant answer to the user's query about Sam Altman's involvement with Microsoft as of November 22, 2023 ... The response is accurate, informative, and directly addresses the user's question". "rating": 5}

Table 6.7: Example of query, LLM response using retrieved passages, and the corresponding LLM rating

for OpenIE-style lexicalized extraction on news chunks, capturing interactions such as emerging events and associations between entities.

KGs for Temporal QA. Temporal QA aims to generate answers for queries with time-specific constraints (Saxena et al., 2021; Mavromatis et al., 2022). KGs provide structured and explainable information that can complement LLMs in answering questions. However, these approaches have limitations in handling evolving entity-specific temporal queries and mostly focus on temporal reasoning tasks (Jia et al., 2024; Saxena et al., 2021). This work builds on these prior methods by incorporating propositions extracted from news streams into the QA process, thereby addressing the temporal and entity-specific challenges.

Retrieval Models. Recent research has focused on retrieval-augmented LLMs, which combine information from external sources with parametric models to improve the factuality of text generation (Guu et al., 2020; Lewis et al., 2020b; Asai et al., 2024). But extensive study on entity-centric temporal RAG remain scarce. For instance, GraphRAG (Edge et al., 2025) builds entity KGs and generates summarizes for closely related entities to support query responses. While effective at capturing global dependencies, these summaries can overlook nuanced, temporally evolving entity-interactions that are critical for reliable temporal QA.

6.7 Summary

We present NEON, a recall-centric information extraction (IE) method to support answering temporal questions by leveraging entity interactions extracted from news streams. By extracting entity-focused, timestamped propositions and temporally retrieving relevant information to augment LLM prompts, the NEON methodology leads to generating more relevant answers, making it a valuable addition to the QA landscape. Overall, the main takeaway from this research is that extracting

OpenIE-style propositions and temporally organizing them enables grounded, faithful answers to telegraphic user queries. This work can be extended by strengthening automated evaluation, expanding the dataset to include more entities and queries, and exploring additional applications of the framework, such as enterprise or personal data integration.

7 Conclusion

7.1 Summary

This thesis presents methods for information extraction (IE) with the goal of achieving high recall: high coverage of extracted facts for a given subject-relation pair. The central question driving the research is: *"How can we extract all the correct structured facts from vast, unstructured sources?"* While traditional IE methods prioritize precision over recall, knowledge-intensive systems demand both. To achieve completeness, IE frameworks should be able to handle *exhaustive extraction*, *ambiguous and boundary cases*, and *long-tail entities and multi-valued relations*. To this end, this thesis explores IE across diverse settings, including web-scale documents, parametric knowledge in large language models (LLMs), and long narrative texts.

When looking at web-scale documents, Chapter 3 formulates the extraction problem as a document coverage prediction task. For this, we develop HERB, a lightweight classifier that acts as an efficient document pre-processor. HERB prioritizes documents by their estimated recall before embarking on exhaustive extraction. Our results demonstrate that, under strict budget constraints and in the relation extraction task as an extrinsic use case, using HERB leads to a higher information yield for the same computational cost. Here, for evaluation, we construct a large-scale ground truth by taking the union of all correct objects associated with a given subject-relation pair. Complementing this, Chapter 6 focuses on OpenIE-style extraction from news streams (web-scale documents characterized by evolving facts). We introduce NEON, a methodology for extracting subject-relation-object triples as propositions. Integrating these propositions for answering temporal questions, in the retrieval-augmented generation (RAG) mode with LLMs, leads to more faithful responses.

With LLMs scaling in size and memorizing factual knowledge, Chapter 4 investigates using LLMs as implicit knowledge bases (KBs). We elicit structured triples via cloze-style prompts, targeting multi-valued relations and prominent subjects—where valid objects can span over both popular (head) and rare (tail) entities. We show that incorporating domain-specific knowledge while crafting prompts improves extraction, but also uncover sparseness and calibration issues. Specifically, LLMs perform reliably on straightforward relations with head entities, yet struggle with multi-valued relations and long-tail entities. This highlights the limits of using LLMs for large-scale KB construction.

Pushing the boundary of context length in RAG mode, Chapter 5 focuses on extracting long lists of object entities from long documents, such as books. In this setting, the relevant cues for extraction are subtle and scattered. We propose L3X, a two-stage framework that combines RAG with iterative LLM prompting for entity list extraction. Unlike standard RAG approaches, L3X integrates pseudo-relevance feedback technique from IR and structured prompting to target relevant passages and maximize recall. Moreover, using the `DoCo` corpus established earlier in Chapter 3, we demonstrate that L3X improves recall not only for narrative texts but also for web-based entity list extraction.

In summary, this thesis moves beyond one-size-fits-all extraction. Through extensive experimentation, we notice that the precision-recall trade-off still persists. Nonetheless, by addressing the challenges of document selection, temporal dynamics, LLM probing, and retrieval depth in long narratives, we lay the groundwork for the next generation of recall-oriented IE systems. Next, Section 7.2 discusses the lessons learned; Section 7.3 outlines the limitations of the proposed frameworks; and finally, Section 7.4 presents opportunities for future research and a broader outlook.

7.2 Lessons Learned

The individual chapters in this thesis are driven by different problem settings, datasets, and modeling choices, but they share a common goal: optimizing IE systems for higher recall without losing too much precision. Looking across these projects, several insights emerge. In this section, we summarize the most important lessons from each of them.

Two-stage extraction is better than direct F1-score optimization for high-recall. We observe that simply optimizing for F1 (harmonic mean of precision and recall) is not sufficient for high-recall IE frameworks. Instead, allowing the model to freely generate a wide range of candidate answers (which will include false positives) and then applying a second-stage precision-oriented scrutinizer/selector for pruning errors helps. This is evident from the results in Chapter 4 and Chapter 5, where we let the LLM produce an unbounded candidate list of answers (which often included more than one ground truth object), and then apply thresholding and other neural methods to recover precision. This way, the system gets a chance to include low-confidence, long-tail facts that are often missed under conservative settings.

Ensemble strategies help to boost recall. Using different starting points and taking a union of all extractions leads to higher recall. For instance, for domain-specific prompts in Chapter 4 and L3X in Chapter 5, we use different prompt templates and multiple retrieval seeds. This is especially useful for covering edge cases with LLM-based generation; if one prompt fails on a particular entity or phrasing, another could succeed. The obvious downside to this is the increased computational cost (further discussed in Section 7.3), but for recall-oriented IE, this is helpful.

Combining explicit and implicit signals improves extraction. Across projects, we find that combining explicit features (pattern matches, lexical cues, paraphrasing tokens, source metadata) with implicit features (neural embeddings, latent representations) consistently yields better results. For instance, in the Herb coverage predictor (Chapter 3), explicit signals like keywords overlap and source attributes are integrated with latent document embeddings when training the classifier. In L3X (Chapter 5), Stage 1 leverages explicit cues like context entities from surrounding passages for prompt contextualization and co-occurring entities for effective retrieval of passages. In Stage 2, the retrieved support passages for creating the predicate classifier are re-ranked with explicit predicate-specific cues.

L3X also incorporates a large-scale alias dictionary for entity disambiguation, enabling more reliable evaluation. Similarly, in Neon (Chapter 6), explicitly marking entities in the prompt text assists LLMs to extract the correct propositions. Overall, the hybrid approach of mixing hand-crafted signals with learned model features prove to be an effective design principle. Nevertheless, named-entity disambiguation remains a challenge across chapters: the correct objects may be generated, but their surface forms do not always match the ground truth (e.g., "Ireland" vs. "Republic of Ireland")

Retrieval helps both efficiency and coverage. Incorporating retrieval-based filtering or ranking of documents prove effective across our methods. For training Herb (Chapter 3), using BM25 as a retrieval step to prioritize informative documents makes exhaustive extraction more efficient. Similarly, in L3X (Chapter 5), using a dense retriever (Contriever) to construct a candidate pool of passages, narrows down where in the long document the answers are likely to appear, thereby improving both efficiency and the focus on recall. Moreover, lexical retrieval performs on par with neural retrieval: in L3X, BM25 achieves retrieval quality similar to Contriever in Stage 1.

Contextualization is required even with LLMs. We find that augmenting prompts with relevant passages and in-context examples (instructions and sample input–output pairs) is crucial for accurate extraction. In L3X (Chapter 5), this is evident from Table 5.6 results: zero-shot prompting has the lowest recall, while few-shot prompting and retrieval-augmented generation achieve higher recall. Larger models and training on web-scale data allows LLMs memorize more knowledge, but it does not eliminate the need for supporting documents during extraction.

Entity popularity skews performance. Across projects, entity popularity, measured based on demographics and web presence, has a direct impact on extraction performance. Head or popular entities, those that are usually well-covered in English-language sources, are much easier to extract facts about. This is because they appear in many documents and contexts, providing the model with abundant cues. In contrast, long-tail or emerging entities consistently prove challenging, especially for multi-valued relations. This insight is reinforced by drilled down of results based on entity popularity (Sections 4.7, 5.6.3, and 6.3.1).

Hallucinations can improve recall. Interestingly, we observe that when the LLM is allowed to generate unbounded candidate answers, even if not grounded in the input prompt, it can help to increase recall. For instance, using L3X (Chapter 5) for the list extraction scenario, the LLM generates plausible entities which are not explicitly mentioned in the retrieved passages, essentially generating from its parametric knowledge. Although this yields some false positives, it also surfaces valid answers. Actually, this behavior is a form of hallucination, but it helps L3X to achieve a higher recall in Stage 1. However, this makes the system less explainable and introduces the risk of unfaithful extractions.

Complete precision and recall remains elusive. Achieving near-100% precision and near-100% recall simultaneously is not possible with current techniques. The trade-off persists, and our work balances it: while we improve recall, precision remains decent but not perfect. In practice, the relative importance of precision versus recall is application-dependent. For instance, in Chapter 4, where the underlying use case is direct knowledge base construction, high precision, and even combining with affordable manual curation, is indispensable. By contrast, in scenarios such as Chapter 5, very high recall at moderate precision levels (e.g., 60 or 80%) is more favorable.

7.3 Limitations

The lessons above paint an optimistic picture of what recall-oriented IE systems can achieve. Each method in this thesis makes simplifying assumptions or design choices, leading to clear caveats. In this section, we summarize the main limitations that emerged during our experiments.

Exact coverage prediction remains difficult. In Chapter 3, HERB performs a binary classification of documents into either informative (1 or high-coverage) or non-informative (0 or low-coverage) class. Predicting the exact coverage of the document, through a variant of HERB as a regressor, did not prove effective and is challenging to train. This is because the distribution of coverage scores across documents for a given subject-relation pair is highly skewed. Hence, we can only predict with a reasonable accuracy if a document has something useful or not, but not exactly how much information it contains for a given task.

Ground-truth incompleteness. A practical limitation across our projects is the construction of ground-truth datasets. We manually or semi-manually curate "all" valid objects for a given subject-relation pair (required for both training and evaluation). This process is time-consuming, does not scale, and can involve ambiguous cases—especially for long-tail entities and temporal scenarios. Consequently, our evaluations can still miss some valid facts (penalizing recall), and our models are limited by the scope of what we consider as valid. This highlights the broader issue that creating truly complete ground-truth is extremely demanding.

Sensitivity towards prompt formulation. In Chapter 4, we rely on discrete or hand-crafted prompts for LLM-based extraction. While discrete prompts are interpretable and controllable for eliciting structured outputs, they are inherently brittle. The output tokens are sensitive to prompt phrasing. Similarly, the choice of in-context examples in few-shot prompting in Chapter 5 (L3X), can vary the model behavior towards the entity list generation task. As a result, although prompt design plays a useful role in our methods, it remains an unsatisfactory long-term solution. Building learnable prompting strategies for these tasks without extensive human intervention remains open for future work.

Higher recall often meant higher computation. A clear limitation of all our recall-oriented approaches is increased computational cost. Especially in Chapter 5, since we use ensemble of prompts, perform iterative prompting, employ multiple retrieval strategies, and devise multi-stage extraction pipelines, this leads to a higher resource consumption on high-computing GPU clusters. When operating at web-scale or in real-time applications, this is a practical bottleneck.

Context length and batch-size limitations. In Chapter 5, where the focus is on extraction from long documents, simply feeding an entire book into a long-context model does not yield good results (see Table 5.18). With open-source LLMs such as Llama-70B, used for list generation within L3X, context length is still limited. So, books are chunked into overlapping passages and augmented into the prompts in the form of batches. However, increasing the batch-size does not linearly improve the performance, and instead shows diminishing returns (see Section 5.6.5). Determining the optimal batch-size and passage ordering for a given book–subject–relation tuple, in order to perform effective multi-passage inference and extraction, remains an open problem.

Minimal model tuning and limits of RAG. Our methods rely on off-the-shelf retrievers (BM25 and Contriever) and LLMs (BERT, GPT-3.5, Llama, and others) without further task-specific tuning for information extraction. A key limitation for potential fine-tuning is the absence of large-scale ground-truth (particularly the corresponding passages for ground-truth objects), and hence we resort to unsupervised techniques. Also, while there is related work on pretraining LLMs to better utilize retrieved passages (Levine et al., 2022; Shi et al., 2024a), these approaches are optimized for a single correct answer. Formulating an effective loss function for recall-oriented tasks is non-trivial.

Smeared-out signals from neural embeddings. Stage 2 in L3X (Chapter 5) uses dense neural embeddings for building the scrutinizer, which does not yield clear benefits. It is difficult to interpret what the embeddings are prioritizing. The cosine similarity scores fail to clearly distinguish relevant from irrelevant passages. This suggests that more work is needed to understand and improve dense retrievers and embedding-based classifiers for IE-specific tasks.

Bias in unified LLM usage. In the Neon methodology (Chapter 6), we used the same LLM both for extracting propositions and for generating the final responses. This can introduce bias or feedback effects, as the model may implicitly favor its own style. Another limitation is the reliance on LLM-as-a-Judge setup to evaluate the responses. Such evaluations are susceptible to positional bias, verbosity bias, and sensitivity to prompt phrasing, which can lead to deviations from task-specific evaluation criteria.

7.4 Outlook

The lessons and limitations of this thesis point to several directions for future work. This includes extending the proposed methods to other languages and beyond our current domains, and rethinking how retrieval, prompting, and modeling should look like in truly recall-first systems. This section outlines the directions that would meaningfully advance high-recall IE in the coming years.

Multilingual and Cross-Lingual IE. All our experiments and datasets are in English, yet a large fraction of web content is in other languages. Adapting our frameworks to multilingual settings poses unique methodological challenges. This is due to the non-uniform performance of underlying NLP components. Since our approaches depend on tokenizers, retrievers, and entity linkers, they are susceptible to error propagation when these components underperform in target languages.

Moreover, cross-lingual evaluation is complicated by inconsistent factual coverage. For example, the English Wikipedia page of *Sunil Chhetri*, an Indian footballer[1], contains substantially more

[1] https://en.wikipedia.org/wiki/Sunil_Chhetri

content than its Russian counterpart[2]. Similarly, the Chinese page[3] of *Jiawei Han* is far more comprehensive than the English version[4], especially for relations such as `membership`. As a result, an IE system might extract a fact from one language that does not exist in another, leading to cross-lingual inconsistencies. One could, for instance, employ a powerful off-the-shelf multilingual LLM to translate documents from a source to a target language. But important nuances can get lost in translation. Future work could explore cross-lingual retrieval and multilingual LLM prompting to ensure that high recall is achievable when relevant information is distributed across languages.

Integration with search and real-time data. Another direction is to integrate our IE methods with search engines or live data feeds to improve recall. Our implemented systems largely assume a static corpus or a frozen LLM. It would be compelling to build a system that, given an information need, can proactively search the web or databases in real time and then apply an IE pipeline to extract missing facts from the results. This could involve iterative search (with query refinement, as in our pseudo-relevance feedback approach in Chapter 5, or similar to parallel work by Jin et al. (2025) combined with IE to handle previously unseen queries. Exploring how such a search-driven IE system could continuously update a knowledge base remains an open challenge.

Optimizing scalability and representations. Despite advances in recall-oriented IE, absolute performance remains constrained by retrieval bottlenecks and limitations of latent representations. Even state-of-the-art dense embedding models struggle to effectively rank evidence as context length increases and signals smear out. Instead of scaling model size, we could develop smaller yet more efficient IE frameworks by training domain-adapted retrievers and fine-tuning LLMs specifically for IE tasks. In particular, exploring multi-stage retrieval (combining dense and sparse methods) and leveraging contrastive learning with synthesized hard negatives to better distinguish true from false positives could be fruitful. By prioritizing energy-aware optimization, it may be possible to achieve robust and cost-effective IE.

Foundation models for IE. Another direction is to build foundation models explicitly trained for IE. Recent efforts aim to unify IE sub-tasks (e.g., named entity recognition, relation extraction, event detection, coreference resolution) within general-purpose, instruction-tuned models (Wang et al., 2023b). However, these models often hallucinate, struggle to handle conflicts between parametric knowledge and retrieval-based inputs, prioritizing precision over recall (Xu et al., 2024a). One could train models specifically for extraction (not for output generation for users) so that models would faithfully adhere to their inference-time context. Insights from our work on long-form extraction and self-supervised pruning could help build reliable, modular IE foundations.

Domain-specific high-recall IE. The presented recall-oriented methods can be applied to other specialized domains such as biomedical information. For example, one could extract all side effects of a drug from medical literature, or all gene–disease associations from research papers. These domains have rich, high-stakes information where recall is critical. However, this would intro-

[2] https://ru.wikipedia.org/wiki/Sunil_Chhetri
[3] https://baike.baidu.com/item/%E9%9F%A9%E5%AE%B6%E7%82%9C/2914641
[4] https://en.wikipedia.org/wiki/Jiawei_Han

duce domain-specific challenges, and most importantly the need for perfect precision. Extending and evaluating the proposed approaches in such domains (and others, like law or finance) is an exciting avenue for future work, which could generalize the lessons from this thesis to a broader scope.

Ultimately, the message of this thesis is simple: to build knowledge-intensive systems that we can trust, we must first ensure that we do not miss the facts that matter.

Appendix

A.1 Details on Chapter 4

Tables A.1–A.7 present the proposed prompt templates for each multi-valued relation.

A.2 Details on Chapter 5

Tables A.8 and A.9 present the predicate-specific query templates for calling the Contriever dense retriever. Tables A.10 to A.17 present the predicate-specific LLM prompt templates. In the zero-shot variant, in-context examples are removed from the prompt templates. Combining the prompt and query templates leads to ensemble mode.

A.3 Details on Chapter 6

Tables A.18–A.20 present the prompt templates for automatic evaluation of responses to user queries.

Prompt Template	optimal k	precision	recall	max-F1	p@1
[X] consists of [MASK].	81.08	42.68	68.58	45.14	31.66
[X] consists of [MASK]	251.54	22.73	26.18	20.08	11.05
[X] consists of [MASK] element.	48.12	49.8	66.64	49.28	37.56
[X] consists of [MASK] element	142.47	33.57	54.88	36.63	13.13
[X] consists of [MASK], which is an element.	33.44	67.84	72.2	64.11	68.00
[X] consists of [MASK], which is an element	39.69	60.18	70.56	57.03	54.27
The chemical compound [X] consists of [MASK].	23.11	50.65	73.09	52.23	42.00
The chemical compound [X] consists of [MASK]	196.56	26.73	27.74	23.52	13.00
The chemical compound [X] consists of [MASK] element.	26.91	47.05	64.6	43.38	33.00
The chemical compound [X] consists of [MASK] element	51.63	36.05	66.4	39.33	15.00
The chemical compound [X] consists of [MASK], which is an element.	21.43	63.34	69.99	59.83	56.50
The chemical compound [X] consists of [MASK], which is an element	28.12	56.23	69.19	51.97	48.50
[X] contains [MASK].	42.52	46.27	75.88	52.78	36.50
[X] contains [MASK] atom	20.13	72.82	78.49	72.45	78.50
[X] is composed of [MASK], which is an element.	23.99	66.23	74.15	64.01	65.50
[X] is composed of [MASK], which is an element	30.2	56.93	71.69	55.02	52.00
[X] is composed of [MASK] atom.	20.82	68.12	79.93	69.18	62.50
[X] is composed of [MASK] atom	21.11	70.26	75.46	67.22	71.00
[X] is composed of [MASK].	58.15	47.92	72.85	50.77	36.00
[X] is composed of [MASK]	172.04	28.28	35.87	26.18	14.75
[MASK] atom is present in [X].	32.04	35.01	79.21	45.23	4.50
[MASK] atom is present in [X]	35.25	33.03	76.98	42.03	5.50
[MASK] element is present in [X].	377.61	8.5	10.58	8.27	1.00
[MASK] element is present in [X]	312.44	12.47	15.98	12.47	2.51
[MASK] is present in [X].	79.54	27.81	66.19	35.53	7.54
[MASK] is present in [X]	111.5	27.42	61.21	33.08	8.25
[X] has [MASK], which is an element.	27.92	69.34	77.11	69.02	68.00
[X] has [MASK], which is an element	32.81	70.32	73.68	67.91	68.84
[X] has [MASK], which is an atom.	20.07	78.75	82.76	78.52	76.00
[X] has [MASK], which is an atom	20.02	74.81	81.77	75.95	74.50
[X] molecule is composed of [MASK], which is an element.	20.5	73.41	78.48	72.04	76.00
[X] molecule is composed of [MASK], which is an element	23.7	65.93	74.53	63.84	62.50
[X] molecule is composed of [MASK] atom.	20.62	70.35	80.6	71.32	69.00
[X] molecule is composed of [MASK] atom	20.77	71.7	77.44	69.94	73.00
[X] molecule is composed of [MASK].	21.23	63.81	80.4	67.24	53.00
[X] molecule is composed of [MASK]	149.31	37.28	34.97	30.58	24.00
[MASK] atom is present in [X] molecule.	23.69	41.1	79.8	50.42	10.50
[MASK] atom is present in [X] molecule	24.7	36.06	78.64	45.96	7.00
[MASK] element is present in [X] molecule.	144.71	19.53	47.54	25.17	1.50
[MASK] element is present in [X] molecule	190.69	20.99	38.21	22.93	4.50
[MASK] is present in [X] molecule.	71.54	31.73	68.47	37.97	10.00
[MASK] is present in [X] molecule	69.52	29.54	72.49	38.01	6.00
The [X] molecule consists of [MASK].	20.48	72.41	81.54	73.33	59.50
The [X] molecule consists of [MASK]	297.46	27.53	19.48	20.14	17.00
The [X] molecule consists of [MASK] element.	31.69	56.43	73.09	57.56	51.50
The [X] molecule consists of [MASK] element	54.96	39.27	64.37	43.4	11.50
The [X] molecule consists of [MASK], which is an element.	20.32	75.89	78.82	73.64	76.00
The [X] molecule consists of [MASK], which is an element	23.21	70.36	76.15	67.98	66.50
[X] molecule has [MASK], which is an element.	20.05	75.17	82.05	75.9	75.50
[X] molecule has [MASK], which is an element	20.1	75.67	81.32	75.68	77.00

Table A.1: Our proposed prompts for (chemical compound, has parts) relation.

Prompt Template	optimal k	precision	recall	max-F1	p@1
[X] shares border with [MASK].	9.3	76.73	77.11	71.29	83.24
[X] shares border with [MASK]	10.92	65.44	70.6	62.02	70.81
[X] shares a border with [MASK].	9.49	78.63	76.59	72.1	83.78
[X] shares a border with [MASK]	11.62	61.24	69.24	58.22	65.95
[X] borders [MASK].	10.37	72.79	74.9	67.98	80.00
[X] borders [MASK]	25.04	25.82	60.28	28.67	12.97
[X] has borders with [MASK].	9.54	79.26	75.14	71.41	82.70
[X] has borders with [MASK]	19.76	49.49	62.74	43.75	48.65
[X] shares border with [MASK], which is a country.	10.01	77.89	77.04	71.49	82.16
[X] shares border with [MASK], which is a country	10.08	78.13	76.62	71.38	81.62
The neighbouring country of [X] is [MASK].	11.07	76.04	74.02	69.13	80.00
The neighbouring country of [X] is [MASK]	74.58	21.43	56.52	28.12	4.32
The neighbouring countries of [X] are [MASK].	46.29	35.97	61.2	39.71	21.08
The neighbouring countries of [X] are [MASK]	149.41	23.07	43.63	24.41	8.65
[X] shares a border with [MASK], which is a country.	10.46	77.38	77.05	71.45	82.16
[X] shares a border with [MASK], which is a country	10.27	78.38	76.07	71.32	83.78
[X] borders [MASK], which is a country.	10.08	78.15	75.92	71.54	82.16
[X] borders [MASK], which is a country	10.03	76.53	76.39	71.04	81.62
[MASK] is a neighbouring country of [X].	10.14	76.35	75.51	70.02	83.24
[MASK] is a neighbouring country of [X]	10.51	73.66	74.97	68.1	80.00
Which country shares border with [X]? Answer: [MASK].	8.74	58.42	71.62	57.49	53.51
Which country shares border with [X]? Answer: [MASK]	119.09	19.56	45.47	22.45	5.41
Which country is near [X]? Answer: [MASK].	8.12	55.89	70.71	56.9	45.95
Which country is near [X]? Answer: [MASK]	89.11	19.21	45.86	22.92	5.95
[MASK], which is a country, is near [X].	9.75	76.37	76.25	70.35	82.16
[MASK], which is a country, is near [X]	9.63	69.4	72.55	64.93	71.89
The country, [MASK], shares border with [X].	9.42	71.85	71.59	64.32	74.59
The country, [MASK], shares border with [X]	11.21	62.09	67.23	56.32	64.32
[MASK] is the bordering country of [X].	10.91	75.9	75.07	69.37	81.62
[MASK] is the bordering country of [X]	12.69	65.9	71.83	61.78	70.27
The country, [MASK], shares border with [X], which is a country.	8.52	71.85	74.25	66.52	75.68
The country, [MASK], shares border with [X], which is a country	8.69	71.83	73.57	66.28	76.76
[MASK], which is a country, shares a border with [X].	11.26	75.08	74.45	68.71	80.00
[MASK], which is a country, shares a border with [X]	11.98	68.41	72.85	64.18	74.05
[MASK], which is a country, borders [X].	11.23	75.63	75.7	69.52	80.54
[MASK], which is a country, borders [X]	11.9	72.8	74.06	67.23	75.68
[MASK], which is a country, has borders with [X].	13.42	72.52	74.88	67.53	77.84
[MASK], which is a country, has borders with [X]	12.77	67.38	71.64	62.24	71.89
The neighbouring country of [X] is [MASK], which is a country.	9.43	78.01	75.52	71.19	83.24
The neighbouring country of [X] is [MASK], which is a country	8.99	76.85	75.96	70.78	83.24
Which country shares border with [X]? The answer is [MASK].	234.62	27.4	28.8	24.76	21.62
Which country shares border with [X]? The answer is [MASK]	477.46	2.52	1.52	1.82	1.08
Which country shares border with [X]? Answer: [MASK], which is a country.	7.36	67.73	73.18	64.68	67.57
Which country shares border with [X]? Answer: [MASK], which is a country	7.92	61.61	73.56	61.44	61.62
[X] and [MASK] share a border.	9.19	80.08	76.7	72.82	83.78
[X] and [MASK] share a border	9.18	78.29	77.21	71.96	84.86
[X] and [MASK] are neighbouring countries.	10.86	73.27	73.9	67.27	77.30
[X] and [MASK] are neighbouring countries	11.32	72.26	73.86	66.7	77.30
[X] and [MASK] are neighbours.	11.08	77.04	75.67	70.76	83.24
[X] and [MASK] are neighbours	8.22	75.98	75.1	70.06	82.70

Table A.2: Our proposed prompts for (country, shares borders) relation.

Prompt Template	optimal k	precision	recall	max-F1	p@1
The official language of [X] is [MASK].	18.18	92.58	80.43	83.54	91.84
The official language of [X] is [MASK]	18.22	90.73	79.66	81.66	88.27
[MASK] is the official language of [X].	18.16	93.09	80.04	83.4	92.35
[MASK] is the official language of [X]	18.2	92.15	80.17	83.1	91.33
[X] has [MASK] as its official language.	18.3	87.26	79.62	79.51	83.16
[X] has [MASK] as its official language	18.43	83.35	79.92	77.36	75.00
The official languages of [X] are [MASK].	18.19	92.08	80.3	83.02	90.31
The official languages of [X] are [MASK]	25.24	29.18	77.04	33.08	16.84
The main language spoken in [X] is [MASK].	18.22	92.43	79.69	83.41	92.35
The main language spoken in [X] is [MASK]	18.42	86.31	79.28	79.09	82.14
People of [X] mostly speak in [MASK], which is a language.	18.2	92.17	80.03	83.26	91.84
People of [X] mostly speak in [MASK], which is a language	18.18	92.31	79.86	83.26	92.86
People of [X] mostly speak in [MASK].	18.13	93.37	79.66	83.41	93.37
People of [X] mostly speak in [MASK]	18.26	88.45	79.75	80.4	84.69
[MASK] is the main spoken language of [X].	18.14	93.22	79.66	83.43	92.35
[MASK] is the main spoken language of [X]	18.16	92.49	79.66	83.05	91.84
[MASK] is spoken in [X].	26.31	66.55	78.05	63.59	60.20
[MASK] is spoken in [X]	26.92	59.87	77.15	58.95	50.51
Language spoken in [X] is [MASK].	18.27	91.67	79.95	82.86	91.33
Language spoken in [X] is [MASK]	19.35	75.56	78.09	70.91	67.86
Languages spoken in [X] are [MASK].	18.95	75.13	78.81	71.35	66.33
Languages spoken in [X] are [MASK]	49.03	12.19	75.08	18.92	0.00
What are the main languages spoken in [X]? Answer: [MASK].	18.44	81.97	80.09	76.73	71.94
What are the main languages spoken in [X]? Answer: [MASK]	119	13.16	61.35	18.35	2.55
What are the official languages of [X]? Answer: [MASK].	18.4	83.09	80.47	77.59	72.45
What are the official languages of [X]? Answer: [MASK]	61.16	14.15	69.61	20.96	1.02
What is the official language of [X]? Answer: [MASK].	18.35	84.53	80.17	78.37	73.98
What is the official language of [X]? Answer: [MASK]	68.94	15.07	69.35	21.67	3.06
Which language is officially spoken in [X]? Answer: [MASK].	18.44	82.86	80.47	77.3	74.49
Which language is officially spoken in [X]? Answer: [MASK]	80.3	14.68	66.10	19.17	4.08
[MASK] is the main language of [X].	18.15	93.22	80.04	83.57	92.35
[MASK] is the main language of [X]	18.16	92.71	79.92	83.32	91.84
In [X], people speak in [MASK].	18.26	91.93	79.86	82.93	92.35
In [X], people speak in [MASK]	19.64	78.75	78.74	72.13	71.94
In [X], people speak in [MASK], which is an official language.	18.22	92.36	79.78	83.18	92.86
In [X], people speak in [MASK], which is an official language	18.23	92.39	79.69	83.18	92.35
In [X], people speak in [MASK], which is a language.	18.21	91.8	79.78	82.9	91.84
In [X], people speak in [MASK], which is a language	18.19	91.89	79.35	82.78	90.82
In [X], people mainly speak in [MASK].	18.14	92.73	79.66	83.15	92.35
In [X], people mainly speak in [MASK]	18.49	87.07	79.16	78.43	83.67
In [X], people mainly speak in [MASK], which is an official language.	18.17	92.46	79.92	82.96	93.37
In [X], people mainly speak in [MASK], which is an official language	18.21	92.35	79.52	83.05	92.35
In [X], people mainly speak in [MASK], which is a language.	18.21	91.78	79.78	82.91	90.82
In [X], people mainly speak in [MASK], which is a language	18.17	92.19	79.35	82.98	91.33
The national language of [X] is [MASK].	18.18	92.62	80.17	83.38	91.33
The national language of [X] is [MASK]	18.37	89.85	78.85	80.39	86.73
[X] is a country and [MASK] is the official language.	18.23	90.15	80.00	81.59	87.76
[X] is a country and [MASK] is the official language	18.28	88.03	80.17	80.49	83.67
[MASK] is the national language of [X].	18.2	92.24	80.43	83.23	91.84
[MASK] is the national language of [X]	18.2	92.24	80.17	83.2	91.84

Table A.3: Our proposed prompts for (country, has official language) relation.

Prompt Template	optimal k	precision	recall	max-F1	p@1
[X] plays [MASK].	83.77	52.59	60.22	49.03	49.00
[X] plays [MASK]	56.31	31.27	66.57	32.52	20.50
[X] plays [MASK] instrument.	17.48	36.88	76.01	42.07	24.00
[X] plays [MASK] instrument	19.72	35.9	73.79	39.48	20.00
[X] plays [MASK] musical instrument.	59.91	16.09	62.72	19.48	5.00
[X] plays [MASK] musical instrument	71.2	11.67	61.89	15.84	2.50
[X] plays [MASK], which is an instrument.	14.12	65.16	74.57	61.24	61.50
[X] plays [MASK], which is an instrument	14.1	67.33	74.74	62.45	66.00
Musician [X] plays [MASK].	15.1	67.36	74.24	61.23	67.50
Musician [X] plays [MASK]	54.72	21.35	66.87	26.93	8.00
Musician [X] plays [MASK] instrument.	30.13	17.17	72.3	23.44	2.50
Musician [X] plays [MASK] instrument	35.04	17.36	68.15	22.46	2.50
Musician [X] plays [MASK], which is an instrument.	14.12	66.99	73.75	61.72	65.50
Musician [X] plays [MASK], which is an instrument	13.81	68.78	72.2	61.11	65.00
[X] played the [MASK].	14.77	63.76	74.19	59.37	63.00
[X] played the [MASK]	18.53	48.28	72.43	48.46	44.50
[X] played the [MASK], which is an instrument.	14.39	59.86	73.18	57.25	58.00
[X] played the [MASK], which is an instrument	14.21	63.68	70.63	56.69	57.50
[MASK], which is an instrument, was played by [X].	15.02	58.45	74.3	55.99	49.50
[MASK], which is an instrument, was played by [X]	14.68	61.6	73.21	56.85	52.00
[X] likes to play the [MASK].	14.29	66.32	70.17	58.19	62.50
[X] likes to play the [MASK]	24.56	42.42	65.56	41.5	26.00
[X] performed on (her|his) [MASK], which is an instrument.	14.08	63.14	72.04	57.15	64.00
[X] performed on (her|his) [MASK], which is an instrument	14.06	65.43	71.42	58.36	67.00
[X] likes to play the [MASK], which is an instrument.	13.83	66.84	69.79	58.34	61.00
[X] likes to play the [MASK], which is an instrument	13.68	68.08	69.25	58.29	63.50
[X] knows to play the [MASK].	13.97	65.89	69.83	58.75	62.00
[X] knows to play [MASK]	486.81	0.40	1.20	0.32	0.00
[X] knows to play the [MASK] instrument.	20.21	26.09	72.71	32.17	10.50
[X] knows to play [MASK] instrument	17.64	47.08	72.52	47.46	35.50
[X] knows to play the [MASK], which is an instrument.	13.96	64.67	70.09	57.21	58.00
[X] knows to play the [MASK], which is an instrument	13.93	64.96	69	56.24	57.50
[X] can play [MASK].	15.03	59.04	74.3	56.56	57.00
[X] can play [MASK]	292.6	9.42	26.31	11.35	2.00
[X] can play [MASK], which is an instrument.	14.16	67.2	73.89	61.7	66.00
[X] can play [MASK] instrument.	19.54	34.77	72.25	37.44	23.00
[X] is noted for playing [MASK] instrument.	32.89	14.19	70.87	19.81	3.00
[X] is noted for playing [MASK] instrument	35.96	16.07	71.73	21.05	1.50
[X] is noted for playing [MASK], which is an instrument.	14.14	66.44	74.48	61.75	64.50
[X] is noted for playing [MASK], which is an instrument	14.18	65.82	73.17	60.11	65.50
[X] is noted for playing [MASK].	17.33	63.09	73.6	58.68	64.50
[X] is noted for playing [MASK]	38.19	18.92	69.68	23.92	5.50
[X] is noted for playing [MASK], which is an instrument.	14.14	66.44	74.48	61.75	64.50
[X] is noted for playing [MASK], which is an instrument	14.18	65.82	73.17	60.11	65.50
[X] practised [MASK], which is an instrument.	14.31	66.18	70.75	58.3	61.50
[X] practised [MASK], which is an instrument	14.38	66.34	71.79	58.92	64.50
[X] taught [MASK], which is an instrument.	14.27	67.04	72.28	60.67	65.00
[X] taught [MASK], which is an instrument	14.1	67.6	72.64	61.16	66.50
[X] performed on (his|her) [MASK], which is an instrument.	14.44	51.89	73.1	51.84	40.50
[X] performed on (his|her) [MASK], which is an instrument	14.01	63.91	71.72	57.67	67.00

Table A.4: Our proposed prompts for (person, plays an instrument) relation.

Prompt Template	optimal k	precision	recall	max-F1	p@1
[X] speaks in [MASK].	2.61	82.13	72.35	72.78	83.50
[X] speaks in [MASK]	5.73	54.6	61.08	48.62	47.50
[X] can speak in [MASK].	2.56	81.63	69.37	70.76	82.50
[X] can speak in [MASK]	7.26	63.35	66.07	57.14	60.50
[X] communicates in [MASK].	2.94	77.04	71.28	69.93	74.50
[X] communicates in [MASK]	6.71	48.41	64.69	47.07	36.50
[X] spoke in [MASK].	2.59	82.62	70.24	71.32	86.00
[X] spoke in [MASK]	7.28	38.36	59.67	39.74	20.00
[X] communicated in [MASK].	3.09	77.38	71.76	69.41	79.50
[X] communicated in [MASK]	13.03	32.09	58.54	35.86	11.00
[X] knows the [MASK] language.	3.18	76.9	68.85	67.49	78.50
[X] knows the [MASK] language	4.04	70.16	70.24	63.91	75.50
[X] learnt [MASK], which is a language.	2.68	79.65	70.03	70.23	81.00
[X] learnt [MASK], which is a language	3.18	74.11	72.93	68.82	79.00
[X] knows [MASK], which is a language.	2.58	82.19	68.29	69.92	88.50
[X] knows [MASK], which is a language	2.54	81.04	68.78	70.12	87.00
Languages spoken by [X] are [MASK].	2.64	78.99	68.52	68.6	86.50
Languages spoken by [X] are [MASK]	19.64	24.57	57.63	29.39	5.50
In which language can [X] speak? Answer: [MASK].	2.45	82.8	65.31	68.18	91.50
In which language can [X] speak? Answer: [MASK]	110.26	19.29	39.12	23.86	1.00
In which language can [X] talk? Answer: [MASK].	2.53	82.43	64.67	67.2	92.50
In which language can [X] talk? Answer: [MASK]	157.13	18.64	37.15	22.08	3.00
In which language can [X] communicates? Answer: [MASK]	142.3	19.28	36.47	22.15	3.00
In which language can [X] communicates? Answer: [MASK]	142.3	19.28	36.47	22.15	3.00
[X] knows to speak in [MASK].	2.6	82.13	69.34	70.88	86.50
[X] knows to speak in [MASK]	18.97	50.01	61.15	45.91	42.50
[X] speaks in [MASK], which is a language.	2.85	78.28	70.92	70.02	82.00
[X] speaks in [MASK], which is a language	3.06	74.5	70.92	68.25	78.00
[X] can speak in [MASK], which is a language.	2.61	81.86	70.11	71.22	85.50
[X] can speak in [MASK], which is a language	2.88	77.43	71.18	69.82	84.50
In [MASK], [X] spoke.	3.12	72.21	62.23	61.27	74.50
In [MASK], [X] spoke	9.81	39.6	57.87	40.93	26.50
In [MASK], which is a language, [X] spoke.	2.83	77.59	69.16	68.24	80.50
In [MASK], which is a language, [X] spoke	2.91	77.5	69.2	68.11	78.50
[X] learned to speak [MASK] fluently.	2.89	78.81	71.55	70.46	80.00
[X] learned to speak [MASK] fluently	2.91	80.46	72.49	71.38	83.50
[X] learned to speak [MASK], which is a language.	2.73	80.59	70.28	70.4	82.50
[X] learned to speak [MASK], which is a language	3.26	75.33	71.25	68.07	80.00
[X] communicates in [MASK], which is a language.	2.44	83.7	67.28	69.82	85.50
[X] communicates in [MASK], which is a language	2.47	82.43	68.62	70.25	80.50
[X] spoke in [MASK], which is a language.	3.17	74.57	69.58	66.67	79.00
[X] spoke in [MASK], which is a language	3.92	67.89	71.21	63.95	73.00
[X] knows to speak in [MASK], which is a language.	2.48	81.9	68.69	70.36	85.00
[X] knows to speak in [MASK], which is a language	2.97	77.49	70.76	69.88	81.00
[X] learned to speak [MASK].	2.99	76.54	70.91	68.95	80.00
[X] learned to speak [MASK]	76.98	18.98	42.24	22.06	3.50
[X] learnt [MASK] language.	4.46	60.35	66.61	56.26	55.00
[X] learnt [MASK] language	4.62	62.27	68.51	58.51	63.00
[X] addressed in [MASK], which is a language.	2.52	81.3	68.71	70.15	82.00
[X] addressed in [MASK], which is a language	2.69	79.85	70.65	70.75	82.00

Table A.5: Our proposed prompts for (person, speaks a language) relation.

Prompt Template	optimal k	precision	recall	max-F1	p@1
[X] is a [MASK] by profession.	30.87	18.52	35.37	20.78	8.00
[X] is a [MASK] by profession	28.39	20.08	34.17	21.23	13.00
[X] is a [MASK].	19.23	35.01	35.98	29.3	40.50
[X] is a [MASK]	20.02	34.14	33.8	26.34	41.50
[X] is a [MASK], which is a profession.	23.15	30.17	36.72	26.81	31.00
[X] is a [MASK], which is a profession	17.87	35.11	31.63	26.43	40.00
[X]'s profession is [MASK].	61.53	13.22	44.71	17.2	3.00
[X]'s profession is [MASK]	111.9	6.14	22.31	6.72	1.50
[X] worked as a [MASK].	25.75	21.29	33.97	22.68	20.50
[X] worked as a [MASK]	16.41	30.66	27.78	24.41	25.50
[X] was a [MASK] for a living.	20.94	37.44	27.59	23.97	48.00
[X] is a [MASK] for a living.	15.66	38.35	30.19	27.14	53.00
[X] is a well-known [MASK].	14.58	39.76	33.36	30.8	59.00
[X] is a well-known [MASK]	11.64	45.35	32.37	33.21	54.50
[X] worked as a [MASK], which is a profession.	23.93	25.12	35.08	24.33	22.50
[X] worked as a [MASK], which is a profession	22.28	25.84	34.95	24.62	25.00
[X] worked as a [MASK] for a living.	29.44	25.35	32.45	21.22	27.00
[X] worked as a [MASK] for a living	27.66	23.66	32.75	21.08	26.00
[X] works as a [MASK].	32.09	17.79	38.41	21.05	19.00
[X] works as a [MASK]	21.62	25.43	31.52	23.02	20.50
What is the profession of [X]? Answer: [MASK].	158.97	2.68	20.45	4.11	0.00
What is the profession of [X]? Answer: [MASK]	216.16	5.51	6.41	3.68	1.50
What did [X] do for a living? Answer: [MASK].	225.71	1.8	8.83	2.33	0.00
What did [X] do for a living? Answer: [MASK]	326.04	3.28	3.27	2.23	1.00
[X] worked as a professional [MASK].	20.84	26.52	36.39	26.9	24.00
[X] worked as a professional [MASK]	39.51	14.9	27.85	15.78	1.00
[X] works as a professional [MASK].	22.99	25.73	38.29	26.45	23.00
[X] works as a professional [MASK]	35.24	16.39	30.64	17.54	2.00
[X] is a professional [MASK].	33.5	19.12	42.3	23.4	2.50
[X] is a professional [MASK]	30.51	15.1	31.41	18	3.50
[X] was a [MASK].	35.97	24.36	28.63	20.46	13.00
[X] was a [MASK]	29.84	31.56	26.78	21.66	33.00
[X] served as a [MASK].	34.84	19.08	28.5	17.56	3.00
[X] served as a [MASK]	30.13	20.31	25.71	17.14	9.00
[X] served as a [MASK], which is a profession.	26.62	25.43	29.37	21.94	13.00
[X] served as a [MASK], which is a profession	35.59	16.22	24.68	16.07	0.00
[X] became a [MASK].	24.79	22.97	30.52	21.71	20.00
[X] became a [MASK]	21.07	28.28	24.12	19.7	26.00
[X] became a [MASK], which is a profession.	24.82	23.87	35.55	24.4	24.50
[X] became a [MASK], which is a profession	23.56	23.95	30.41	21.78	19.00
[X] is an [MASK].	6.75	63.21	14.61	19.86	54.50
[X] is an [MASK]	14.35	26.24	13.94	15.54	11.00
[X] was an [MASK].	6.71	61.71	13.59	18.83	50.00
[X] was an [MASK]	20.49	31.84	13.04	14.65	18.00
[X] was a professional [MASK].	30.19	17.73	36.75	21.38	4.50
[X] was a professional [MASK]	51.69	12.68	29.00	14.65	0.50
[X] joined as a [MASK], which is a profession.	20.01	33.47	31.79	26.26	36.50
[X] joined as a [MASK], which is a profession	17.54	27.09	29.91	25.06	9.50
[X] joined as a [MASK].	21.72	23.88	31.06	24.16	8.00
[X] joined as a [MASK]	26.56	19.02	30.65	20.96	0.00

Table A.6: Our proposed prompts for (person, has an occupation) relation.

Prompt Template	optimal k	precision	recall	max-F1	p@1
[X] shares border with [MASK].	285.62	35.78	21.85	25.18	32.50
[X] shares border with [MASK]	291.63	25.94	19.15	19.14	26.50
[X] shares a border with [MASK].	285.64	36.08	22.02	25.39	32.50
[X] shares a border with [MASK]	296.5	22.06	18.2	17.32	20.00
[X] borders [MASK].	285.81	35.48	21.51	24.63	34.00
[X] borders [MASK]	316.65	9.03	16.07	9.48	2.50
[X] has borders with [MASK].	286.25	32.6	21.67	24.1	30.00
[X] has borders with [MASK]	306.04	10.59	16.58	10.83	3.50
[X] shares border with [MASK], which is a [Y].	285.38	37.32	21.48	25.21	35.50
[X] shares border with [MASK], which is a [Y]	285.53	36.7	21.8	25.16	34.50
The neighbouring [Y] of [X] is [MASK].	290.67	32.64	20.87	23.52	32.50
The neighbouring [Y] of [X] is [MASK]	371.29	6.00	8.67	6.08	0.50
The [X] [Y] shares border with [MASK].	285.53	35.92	21.6	25.06	32.00
The [X] [Y] shares border with [MASK]	293.25	14.16	17.2	12.97	6.50
[X] shares a border with [MASK], which is a [Y].	285.4	37.27	21.6	25.3	35.00
[X] shares a border with [MASK], which is a [Y]	285.56	36.81	21.77	25.21	35.50
[X] borders [MASK], which is a [Y].	285.51	36.36	22.08	25.1	35.00
[X] borders [MASK], which is a [Y]	285.56	36.13	22.08	25.15	34.00
[MASK] is a neighbouring [Y] of [X].	285.45	35.96	21.79	25.04	34.50
[MASK] is a neighbouring [Y] of [X]	285.51	34.73	21.43	24.52	31.50
Which [Y] shares border with [X]? Answer: [MASK].	293.96	23.81	20.94	20.63	25.00
Which [Y] shares border with [X]? Answer: [MASK]	429.81	6.28	5.16	4.78	3.00
Which [Y] is near [X]? Answer: [MASK].	286.75	23.85	20.66	20.44	20.00
Which [Y] is near [X]? Answer: [MASK]	432.43	5.58	6.21	4.67	1.50
[MASK], which is a [Y], is near [X].	285.42	37.09	21.85	25.39	35.50
[MASK], which is a [Y], is near [X]	286.12	29.84	21.02	22.02	29.50
The [Y], [MASK], shares border with [X].	285.78	32.22	21.64	24.1	31.50
The [Y], [MASK], shares border with [X]	286.05	28.00	21.31	22.26	26.50
[MASK] is the bordering [Y] of [X].	285.41	37.06	21.94	25.43	35.50
[MASK] is the bordering [Y] of [X]	285.54	35.88	21.9	24.69	36.50
The [Y], [MASK], shares border with [X], which is a [Y].	285.61	34.00	21.25	24.23	34.00
The [Y], [MASK], shares border with [X], which is a [Y]	285.64	33.48	21.88	24.29	34.50
[MASK], which is a [Y], shares a border with [X].	285.37	37.96	21.86	25.49	37.00
[MASK], which is a [Y], shares a border with [X]	285.41	37.29	21.5	25.1	36.50
[MASK], which is a [Y], borders [X].	285.35	38.46	21.8	25.65	37.50
[MASK], which is a [Y], borders [X]	285.46	36.25	21.46	24.68	36.00
[MASK], which is a [Y], has borders with [X].	285.37	37.9	21.78	25.4	37.00
[MASK], which is a [Y], has borders with [X]	285.41	36.57	21.07	24.57	36.00
The neighbouring [Y] of [X] is [MASK], which is a [Y].	285.46	36.14	21.78	24.75	35.50
The neighbouring [Y] of [X] is [MASK], which is a [Y]	285.68	33.13	21.73	23.75	32.50
Which [Y] shares border with [X]? The answer is [MASK].	407.63	8.70	6.99	6.63	5.50
Which [Y] shares border with [X]? The answer is [MASK]	499.0	0.00	0.00	0.00	0.00
Which [Y] shares border with [X]? Answer: [MASK], which is a [Y].	285.71	32.77	21.72	23.74	30.50
Which [Y] shares border with [X]? Answer: [MASK], which is a [Y]	291.11	26.46	20.07	20.93	24.00
[X] and [MASK] share a border.	285.49	37.49	22.3	25.69	37.50
[X] and [MASK] share a border	285.49	37.54	22.3	25.71	37.00
The [X] [Y] shares border with [MASK], which is a [Y].	285.43	37.26	21.72	25.36	36.00
The [X] [Y] shares border with [MASK], which is a [Y]	285.51	37.11	21.55	25.22	35.50
[X] and [MASK] are neighbours.	288.2	33.55	21.78	23.96	33.00
[X] and [MASK] are neighbours	290.68	33.77	21.68	23.92	33.50

Table A.7: Our proposed prompts for (state, shares border) relation.

ID	Relation	Template
1	family	Who are the family members of ##subj##, including parent, father, mother, son, daughter, sibling, brother, sister, child, aunt, uncle, auntie, cousin, nephew, niece, grandfather, grandmother, grandson, granddaughter, in-laws, stepfather, stepmother, stepson, stepdaughter, stepbrother, stepsister, half-brother, half-sister, godfather, godmother?
2	family	family members of ##subj##
3	family	Who are the family members of ##subj##?
4	family	Who are the relatives of ##subj##, including parents, children, siblings, aunt, uncle, cousin, nephew, niece, grandparents, grandchildren, in-laws, stepparents, stepchildren, step-siblings, half-siblings, godparents?
5	family	Who are the relatives of ##subj##?
1	parent	Who are the parents, including adopted parents and foster parents, of ##subj##?
2	parent	parents of ##subj##
3	parent	father and mother of ##subj##
4	parent	Who are the father and mother of ##subj##?
5	parent	Who are the parents of ##subj##?
5	sibling	Who are the siblings of ##subj##?
2	sibling	siblings of ##subj##
3	sibling	Who are the brothers and sisters of ##subj##?
4	sibling	brothers and sisters of ##subj##
1	sibling	Who are the siblings, including stepbrother, stepsister, half-brother and half-sister, of ##subj##?
2	children	Who are the children of ##subj##?
1	children	Who are the children, including step, adopted and foster sons and daughters, of ##subj##?
3	children	##subj## children child son daughter
4	children	##subj## children child baby kid
5	children	##subj## son daughter

Table A.8: Predicate-specific retriever query templates for the predicates family, parent, sibling, children. All five templates per predicate are used in the ensemble mode, while only the first one is used in single prompt mode.

ID	Relation	Template
1	friend	Who are the friends, including companion, pal, supporter, buddy, ally, bestie, and follower, of ##subj##?
2	friend	friends of ##subj##
3	friend	Who are the friends or supporters of ##subj##?
4	friend	friends or supporters of ##subj##
5	friend	Who are the friends of ##subj##?
1	opponent	Who are the enemies, including rival, attacker, opponent, foe, nemesis and critic, of ##subj##?
2	opponent	enemies of ##subj##
3	opponent	Who are the enemies or rivals of ##subj##?
4	opponent	enemies or rivals of ##subj##
5	opponent	Who are the enemies of ##subj##?
1	hasMember	Who are the members, including teammates, partners and followers, of ##subj##?
2	hasMember	Who are part of ##subj##?
3	hasMember	##subj## member teammate group part of champions followers belongs member
4	hasMember	##subj## member group family part of
5	hasMember	##subj## member teammate group
1	placeHasPerson	Who lived, travelled, go for a vacation or visited ##subj##?
2	placeHasPerson	Who all travelled or go for a vacation to ##subj##?
3	placeHasPerson	Who resided in ##subj##?
4	placeHasPerson	names of people living, traveling, residing or visting ##subj##
5	placeHasPerson	people in ##subj##

Table A.9: Predicate-specific retriever query templates for the predicates friend, opponent, hasMember, placeHasPerson. All five templates per predicate are used in the ensemble mode, while only the first one is used in single prompt mode.

System:- You are a knowledge base. Generate the complete list of names (Objects) who are parents, including step parents, of the specified person in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: A Promised Land, Subject: Barack Obama, Relation: parent Output: [Barack Obama Senior, Stanley Ann Dunham] Input: Book: The Fellowship of the Ring, Subject: Frodo Baggins, Relation: parent Output: [Drogo Baggins, Primula Brandybuck]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: parent, Passages: {T}

Table A.10: Prompt template for `parent` relation (placeholders in curly brackets).

System:- You are a knowledge base. Generate the complete list of names (Objects) who are children of the specified person in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: A Promised Land, Subject: Barack Obama, Relation: children Output: [Sasha Obama, Malia Ann Obama] Input: Book: Twilight series, Subject: Carlisle Cullen, Relation: children Output: [Edward Cullen, Alice Cullen, Jasper Whitlock Hale, Emmett Cullen, Rosalie Hale]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: children, Passages: {T}

Table A.11: Prompt template for `child` relation (placeholders in curly brackets).

System:- You are a knowledge base. Generate the complete list of names (Objects) who are friends, including supporters, of the specified person in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: Percy Jackson & The Olympians, Subject: Percy Jackson, Relation: friend Output: [Blackjack, Mrs. OLeary, Grover Underwood, Jason Grace, Thalia Grace, Nico di Angelo, Chiron, Clarisse La Rue, Zoë Nightshade, Charles Beckendorf, Silena Beauregard, Bianca di Angelo, Hazel Levesque, Reyna Ramírez-Arellano, Dakota, Leo Valdez, Piper McLean, Gleeson Hedge] Input: Book: To Kill a Mockingbird, Subject: Scout Finch, Relation: friend Output: [Jem Finch, Dill Harris, Atticus Finch, Calpurnia]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: friend, Passages: {T}

Table A.12: Prompt template for `friend` relation (placeholders in curly brackets).

System:- You are a knowledge base. Generate the complete list of names (Objects) who are members of, part of, or associated with the specified group, team, or organization in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: Dune series, Subject: Atreides Empire, Relation: hasMember Output: [Paul Atreides, Irulan Corrino, Alia Atreides, Leto Atreides II] Input: Book: Twilight series, Subject: Vampires, Relation: hasMember Output: [Edward Cullen, Alice Cullen, Carlisle Cullen, Esme Cullen, Jasper Whitlock Hale, Emmett Cullen, Rosalie Hale]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: hasMember, Passages: {T}

Table A.13: Prompt template for `hasMember` relation (placeholders in curly brackets).

System:- You are a knowledge base. Generate the complete list of names (Objects) of people who appear, live, or travel to the specified location in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: Malibu Rising, Subject: Beverly Hills, Relation: placeHasPerson Output: [Brandon Randall] Input: Book: The Void Trilogy, Subject: Hanko, Relation: placeHasPerson Output: [Aaron, Corrie lyn, Inigo]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: placeHasPerson, Passages: {T}

Table A.14: Prompt template for `placeHasPerson` relation (placeholders in curly brackets).

System:- You are a knowledge base. Generate the complete list of names (Objects) of people who are siblings, including step and half siblings, of the specified person in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: The Princess Diaries, Subject: Lilly Moscovitz, Relation: sibling Output: [Michael Moscovitz] Input: Book: Percy Jackson & the Olympians, Subject: Chiron, Relation: sibling Output: [Dolops, Hades, Poseidon, Zeus, Aphros, Bythos]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: sibling, Passages: {T}

Table A.15: Prompt template for `sibling` relation (placeholders in curly brackets).

System:- You are a knowledge base. Generate the complete list of names (Objects) who are enemies or opponents of the specified person in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: The Chronicles of Narnia, Subject: Aslan, Relation: opponent Output: [The White Witch, The Dark Island, Lady of the Green Kirtle] Input: Book: Percy Jackson & The Olympians, Subject: Percy Jackson, Relation: opponent Output: [Luke Castellan, Octavian]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: opponent, Passages: {T}

Table A.16: Prompt template for `opponent` relation (placeholders in curly brackets).

System:- You are a knowledge base. Generate the complete list of names (Objects) who are family members, including relatives, of the specified person in the given book. List the names one after the other, separated by commas.
Few-Shot examples:- Input: Book: Twilight series, Subject: Bella Swan, Relation: family Output: [Charlie Swan, Renée Dwyer, Phil Dwyer, Carlisle Cullen, Esme Cullen, Alice Cullen, Rosalie Hale, Emmett Cullen, Jasper Hale] Input: Book: Percy Jackson & the Olympians, Subject: Percy Jackson, Relation: family Output: [Poseidon, Sally Jackson, Annabeth Chase, Gabe Ugliano, Paul Blofis, Amphitrite, Estelle Blofis, Procrustes, Polyphemus, Triton, Arion, Antaeus, Pegasus, Chrysaor, Chrysomallus, Cercyon, Sciron, Theseus, Bellerophon, Neleus, Halirrhothius, Tyson, Rhode, Kymopoleia, Despoina, Charybdis, Several Cyclops, Zeus, Chiron, Hades, Hestia, Demeter, Hera, Bianca di Angelo, Nico di Angelo, Thalia Grace, Jason Grace, Kronos, Rhea, Jim Jackson, Estelle Jackson, Rich Jackson, Frank Zhang]
User:- Use the attached passages from the book. Book:{B}, Subject:{S}, Relation: family, Passages: {T}

Table A.17: Prompt template for `family` relation (placeholders in curly brackets).

Task description:
You are presented with a user query and an AI assistant's response. The query is focused on a specific entity, pertains to the news domain, and is date-stamped. Your task is to evaluate the AI assistant's response for its usefulness, using a 3-point Likert scale. The criteria for rating are detailed below.
Helpfulness criterion:
Rating 2: The response is very helpful and provides the information expected for the user query. It includes mentions of additional named entities (such as people, locations, events, etc.) beyond the primary entity in the query and aligns completely with user's intent.
Rating 1: The response is somewhat helpful but fails to fully provide the information expected for the user's query. It can nevertheless serve to continue the conversation with the user or provides pointers to where the information can be found.
Rating 0: The response is not helpful and provides no information for the query.
Output format:
The output should be the following JSON format:
{"rating": <numerical_rating>, "reason": <short_reasoning>}, mentioning the numerical rating, as well as a short and concise reasoning for the helpfulness rating.
Examples: {in-context examples}
Input to be rated user query: {question} AI assistant's response: {response}

Table A.18: Prompt Template for Automatic Evaluation using Few-shot Learning on Helpfulness Criterion.

Task description:
You are presented with a user query and an AI assistant's response. The query is focused on a specific entity, pertains to the news domain, and is date-stamped. Your task is to evaluate the AI assistant's response for its relevance, using a 3-point Likert scale. The criteria for rating are detailed below.
Relevance criterion:
Rating 2: The response is completely relevant with accurate details and provides the information for the query date.
Rating 1: The response contains a mix of relevant and irrelevant details. The response contains some relevant information upto the specified date, and is more or less aligned with the user's intent.
Rating 0: The response is incorrect and provides no information for the query date.
Output format:
The output should be the following JSON format:
{"rating": <numerical_rating>, "reason": <short_reasoning>}, mentioning the numerical rating, as well as a short and concise reasoning for the relevance rating.
Examples: {in-context examples}
Input to be rated user query: {question} AI assistant's response: {response}

Table A.19: Prompt Template for Automatic Evaluation using Few-shot Learning on Helpfulness Criterion.

Task description: You are presented with a user query and an AI assistant's response. The query is focused on a specific entity, pertains to the news domain, and is date-stamped. Your task is to evaluate the AI assistant's response for its reliability, using a 3-point Likert scale. The criteria for rating are detailed below. ## **Faithfulness criterion:** Rating 2: The response is perfectly reliable and grounded based on the supporting passages given below. All the information from the supporting passages is used in the response to answer the user query. Rating 1: The response partially uses the supporting passages given below but has additional information which may be incorrect or unreliable. Rating 0: The response is completely unreliable and does not depend on the supporting passages. ## **Output format:** The output should be the following JSON format: {"rating": <numerical_rating>, "reason": <short_reasoning>}, mentioning the numerical rating, as well as a short and concise reasoning for the helpfulness rating. ## **Examples:** {in-context examples} ## **Input to be rated user query:** {question} Supporting passages: {passages} AI assistant's response: {response}

Table A.20: Prompt templates for automated evaluation using few-shot learning.

List of Figures

List of Tables

List of Algorithm

Bibliography

Eugene Agichtein and Luis Gravano. *Snowball*: extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries, June 2-7, 2000, San Antonio, TX, USA*, pages 85–94. ACM, 2000. doi: 10.1145/336597.336644. URL https://doi.org/10.1145/336597.336644.

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. Large language models are few-shot clinical information extractors. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.130. URL https://aclanthology.org/2022.emnlp-main.130/.

David Ahn. The stages of event extraction. In Branimir Boguraev, Rafael Muñoz, and James Pustejovsky, editors, *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia, July 2006. Association for Computational Linguistics. URL https://aclanthology.org/W06-0901/.

Ekin Akyurek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. Towards tracing knowledge in language models back to the training data. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2429–2446, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.180. URL https://aclanthology.org/2022.findings-emnlp.180/.

Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L. Opdahl, and Csaba Veres. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881, 2020. doi: 10.1109/ACCESS.2020.2973928. URL https://doi.org/10.1109/ACCESS.2020.2973928.

Dimitrios Alivanistos, Selene Baez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. Prompting as probing: Using language models for knowledge base construction. In Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski, editors, *Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022 co-located with the 21st International Semantic Web Conference (ISWC2022), Virtual Event, Hanghzou, China, October 2022*, volume 3274 of *CEUR Workshop Proceedings*, pages 11–34. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3274/paper2.pdf.

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona T. Diab, and Marjan Ghazvininejad. A review on language models as knowledge bases. *CoRR*, abs/2204.06031, 2022. doi: 10.48550/ARXIV.2204.06031. URL https://doi.org/10.48550/arXiv.2204.06031.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1034. URL https://aclanthology.org/P15-1034/.

Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Negative statements considered useful. *Journal of Web Semantics*, 71:100661, 2021. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2021.100661.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. Retrieval-based language models and applications. In Yun-Nung (Vivian) Chen, Margot Margot, and Siva Reddy, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-tutorials.6. URL https://aclanthology.org/2023.acl-tutorials.6/.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2007. doi: 10.1007/978-3-540-76298-0_52. URL https://doi.org/10.1007/978-3-540-76298-0_52.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Vevake Balaraman, Simon Razniewski, and Werner Nutt. Recoin: Relative completeness in wikidata. In Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, editors, *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1787–1792. ACM, 2018. doi: 10.1145/3184558.3191641. URL https://doi.org/10.1145/3184558.3191641.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1279. URL https://aclanthology.org/P19-1279/.

Krisztian Balog. Entity-oriented search. In *The Information Retrieval Series*. Springer Charm, 2018. doi: 10.1007/978-3-319-93935-3. URL https://link.springer.com/book/10.1007/978-3-319-93935-3.

David Bamman, Sejal Popat, and Sheng Shen. An annotated dataset of literary entities. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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)*, pages 2138–2144, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1220. URL https://aclanthology.org/N19-1220/.

David Bamman, Olivia Lewke, and Anya Mansoor. An annotated dataset of coreference in English literature. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 44–54, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.6/.

David Bamman, Kent K. Chang, Li Lucy, and Naitian Zhou. On classification with large language models in cultural analytics. In *Proceedings of the Computational Humanities Research Conference 2024, Aarhus, Denmark, December 4-6, 2024*, volume 3834 of *CEUR Workshop Proceedings*, pages 494–527. CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3834/paper119.pdf.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. URL https://arxiv.org/abs/2004.05150.

Klaus Berberich, Srikanta J. Bedathur, Omar Alonso, and Gerhard Weikum. A language modeling approach for temporal information needs. In Cathal Gurrin, Yulan He, Gabriella Kazai, Udo Kruschwitz, Suzanne Little, Thomas Roelleke, Stefan M. Rüger, and Keith van Rijsbergen, editors, *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings*, volume 5993 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 2010. doi: 10.1007/978-3-642-12275-0_5. URL https://doi.org/10.1007/978-3-642-12275-0_5.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. Unlimiformer: Long-range transformers with unlimited length input. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing*

Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6f9806a5adc72b5b834b27e4c7c0df9b-Abstract-Conference.html.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. CaRB: A crowdsourced benchmark for open IE. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1651. URL https://aclanthology.org/D19-1651/.

David M. Blei, Andrew Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. URL https://dl.acm.org/doi/10.5555/944919.944937.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008. doi: 10.1145/1376616.1376746. URL https://doi.org/10.1145/1376616.1376746.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. Inducing relational knowledge from BERT. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7456–7463. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6242. URL https://doi.org/10.1609/aaai.v34i05.6242.

Sergey Brin. Extracting patterns and relations from the world wide web. In Paolo Atzeni, Alberto O. Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases, International Workshop WebDB'98, Valencia, Spain, March 27-28, 1998, Selected Papers*, volume 1590 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 1998. doi: 10.1007/10704656_11. URL https://doi.org/10.1007/10704656_11.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Networks*, 30(1-7):107–117, 1998. doi: 10.1016/S0169-7552(98)00110-X. URL https://doi.org/10.1016/S0169-7552(98)00110-X.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Razvan Bunescu and Marius Paşca. Using encyclopedic knowledge for named entity disambiguation. In Diana McCarthy and Shuly Wintner, editors, *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://aclanthology.org/E06-1002/.

Deng Cai, Yan Wang, Lemao Liu, and Shuming Shi. Recent advances in retrieval-augmented text generation. In Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3417–3419. ACM, 2022. doi: 10.1145/3477495.3532682. URL https://doi.org/10.1145/3477495.3532682. Tutorial materials at "https://jcyk.github.io/RetGenTutorial/".

Mary Elaine Califf and Raymond J. Mooney. Relational learning of pattern-match rules for information extraction. In *CoNLL97: Computational Natural Language Learning*, 1997. URL https://aclanthology.org/W97-1002/.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=5k8F6UU39V.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In Maria Fox and David

Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, pages 1306–1313. AAAI Press, 2010. doi: 10.1609/AAAI.V24I1.7519. URL https://doi.org/10.1609/aaai.v24i1.7519.

Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, 2012. doi: 10.1145/2071389.2071390. URL https://doi.org/10.1145/2071389.2071390.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. Cross-document coreference resolution over predicted mentions. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.453. URL https://aclanthology.org/2021.findings-acl.453/.

Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Tong Xiao, and Jingbo Zhu. Efficient prompting methods for large language models: A survey. *CoRR*, abs/2404.01077, 2024a. doi: 10.48550/ARXIV.2404.01077. URL https://doi.org/10.48550/arXiv.2404.01077.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.453. URL https://aclanthology.org/2023.emnlp-main.453/.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Booookscore: A systematic exploration of book-length summarization in the era of llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=7Ttk3RzDeu.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171/.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *CoRR*, abs/2306.15595, 2023. doi: 10.48550/ARXIV.2306.15595. URL https://doi.org/10.48550/arXiv.2306.15595.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. Seeing things from a different angle:discovering diverse perspectives about claims. In Jill Burstein, Christy

Doran, and Thamar Solorio, editors, *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)*, pages 542–557, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1053. URL https://aclanthology.org/N19-1053/.

Wenhu Chen, Xinyi Wang, and William Yang Wang. A dataset for answering time-sensitive questions. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/1f0e3dad99908345f7439f8ffabdffc4-Abstract-round2.html.

Xiang Chen, Lei Li, Ningyu Zhang, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Relation extraction as open-book examination: Retrieval-enhanced prompt tuning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2443–2448, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531746. URL https://doi.org/10.1145/3477495.3531746.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2778–2788. ACM, 2022b. doi: 10.1145/3485447.3511998. URL https://doi.org/10.1145/3485447.3511998.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1017. URL https://aclanthology.org/P15-1017/.

Yulong Chen, Yang Liu, Li Dong, Shuohang Wang, Chenguang Zhu, Michael Zeng, and Yue Zhang. AdaPrompt: Adaptive model training for prompt-based NLP. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6057–6068, Abu Dhabi, United Arab Emirates, December 2022c. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.448. URL https://aclanthology.org/2022.findings-emnlp.448/.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. Factool: Factuality detection in generative AI - A tool augmented framework for multi-task and multi-domain scenarios. *CoRR*, abs/2307.13528, 2023. doi: 10.48550/ARXIV.2307.13528. URL https://doi.org/10.48550/arXiv.2307.13528.

Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.870. URL https://aclanthology.org/2023.acl-long.870/.

Jason P.C. Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016. doi: 10.1162/tacl_a_00104. URL https://aclanthology.org/Q16-1026/.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. Explainable conversational question answering over heterogeneous sources via iterative graph neural networks. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete, editors, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 643–653. ACM, 2023. doi: 10.1145/3539618.3591682. URL https://doi.org/10.1145/3539618.3591682.

Rudi Cilibrasi and Paul M. B. Vitányi. The google similarity distance. *CoRR*, abs/cs/0412098, 2004. URL http://arxiv.org/abs/cs/0412098.

Kevin Clark and Christopher D. Manning. Entity-centric coreference resolution with model stacking. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1136. URL https://aclanthology.org/P15-1136/.

Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1245. URL https://aclanthology.org/D16-1245/.

Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. Crawling the internal knowledge-base of language models. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1856–1869, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.139. URL https://aclanthology.org/2023.findings-eacl.139/.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit S. Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, Krishna Haridasan, Ahmed Omran, Nikunj Saunshi, Dara Bahri, Gaurav Mishra, Eric Chu, Toby Boyd, Brad Hekman, Aaron Parisi, Chaoyi Zhang, Kornraphop Kawintiranon, Tania Bedrax-Weiss, Oliver Wang, Ya Xu, Ollie Purkiss, Uri Mendlovic, Ilaï Deutel, Nam Nguyen, Adam Langley, Flip Korn, Lucia Rossazza, Alexandre Ramé, Sagar Waghmare, Helen Miller, Nathan Byrd, Ashrith Sheshan, Raia Hadsell Sangnie Bhardwaj, Pawel Janus, Tero Rissa, Dan Horgan, Sharon Silver, Ayzaan Wahid, Sergey Brin, Yves Raimond, Klemen Kloboves, Cindy Wang, Nitesh Bharadwaj Gundavarapu, Ilia Shumailov, Bo Wang, Mantas Pajarskas, Joe Heyward, Martin Nikoltchev, Maciej Kula, Hao Zhou, Zachary Garrett, Sushant Kafle, Sercan Arik, Ankita Goel, Mingyao Yang, Jiho Park, Koji Kojima, Parsa Mahmoudieh, Koray Kavukcuoglu, Grace Chen, Doug Fritz, Anton Bulyenov, Sudeshna Roy, Dimitris Paparas, Hadar Shemtov, Bo-Juen Chen, Robin Strudel, David Reitter, Aurko Roy, Andrey Vlasov, Changwan Ryu, Chas Leichner, Haichuan Yang, Zelda Mariet, Denis Vnukov, Tim Sohn, Amy Stuart, Wei Liang, Minmin Chen, Praynaa Rawlani, Christy Koh, JD Co-Reyes, Guangda Lai, Praseem Banzal, Dimitrios Vytiniotis, Jieru Mei, and Mu Cai. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261, 2025. doi: 10.48550/ARXIV.2507.06261. URL https://doi.org/10.48550/arXiv.2507.06261.

Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In Jason Eisner, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/D07-1074/.

Silviu Cucerzan. Name entities made obvious: the participation in the erd 2014 evaluation. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, page 95–100, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330237. doi: 10.1145/2633211.2634360. URL https://doi.org/10.1145/2633211.2634360.

Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2065. URL https://aclanthology.org/P18-2065/.

Zhuyun Dai and Jamie Callan. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988, New York, NY, USA, 2019. Association for Computing Machinery. doi: 10.1145/3331184.3331303.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

Fariz Darari, Werner Nutt, Giuseppe Pirrò, and Simon Razniewski. Completeness statements about RDF data sources and their use for query answering. In *The Semantic Web – ISWC 2013*, pages 66–83. Springer Berlin Heidelberg, 2013.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290, 2022. doi: 10.1162/tacl_a_00460. URL https://aclanthology.org/2022.tacl-1.16/.

Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 355–366, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320351. doi: 10.1145/2488388.2488420. URL https://doi.org/10.1145/2488388.2488420.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.222. URL https://aclanthology.org/2022.emnlp-main.222/.

Pascal Denis and Jason Baldridge. A ranking approach to pronoun resolution. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 1588–1593, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.21. URL https://aclanthology.org/2020.emnlp-main.21/.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022. doi: 10.1162/tacl_a_00459. URL https://aclanthology.org/2022.tacl-1.15/.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva, editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL https://aclanthology.org/L04-1011/.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL https://aclanthology.org/2021.emnlp-main.98/.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.64. URL https://aclanthology.org/2024.emnlp-main.64/.

Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610, 2014. URL http://www.cs.cmu.edu/~nlao/publication/2014.kdd.pdf. Evgeniy GabrilovichWilko HornNi LaoKevin MurphyThomas StrohmannShaohua SunWei ZhangGeremy Heitz.

Cícero dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1061. URL https://aclanthology.org/P15-1061/.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In Chu-Ren Huang and Dan Jurafsky, editors, *Proceedings of the 23rd*

International Conference on Computational Linguistics (Coling 2010), pages 277–285, Beijing, China, August 2010. Coling 2010 Organizing Committee. URL https://aclanthology.org/C10-1032/.

Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.49. URL https://aclanthology.org/2020.emnlp-main.49/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cris tian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Kenneth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Ma hesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Niko lay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stéphane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Vir ginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm'an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthias Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Mun ish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navy ata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Kumar Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2006–2013. IOS Press, 2020. doi: 10.3233/FAIA200321. URL https://doi.org/10.3233/FAIA200321.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *CoRR*, abs/2404.16130, 2025. URL https://arxiv.org/abs/2404.16130.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410. URL https://aclanthology.org/2021.tacl-1.60/.

Joe Ellis, Jeremy Getman, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M. Strassel. Overview of linguistic resources for the TAC KBP 2016 evaluations: Methodologies and results. In *Proceedings of the 2016 Text Analysis Conference, TAC 2016, Gaithersburg, Maryland, USA, November 14-15, 2016*. NIST, 2016. URL https://tac.nist.gov/publications/2016/additional.papers/TAC2016.KBP_resources_overview.proceedings.pdf.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/632.html.

Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020. doi: 10.1162/tacl_a_00298. URL https://aclanthology.org/2020.tacl-1.3/.

Oren Etzioni, Michael J. Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall: (preliminary results). In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 100–110. ACM, 2004. doi: 10.1145/988672.988687. URL https://doi.org/10.1145/988672.988687.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In Regina Barzilay and Mark Johnson, editors, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1142/.

Meihao Fan, Xiaoyue Han, Ju Fan, Chengliang Chai, Nan Tang, Guoliang Li, and Xiaoyong Du. Cost-effective in-context learning for entity resolution: A design space exploration. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, pages 3696–3709. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00284. URL https://doi.org/10.1109/ICDE60146.2024.00284.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. Lasuie: unifying information extraction with latent adaptive structure-aware generative language model. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971. doi: 10.1037/h0031619.

Rudolf Flesch and Alan J. Gould. *The Art of Readable Writing*, volume 8. Harper New York, 1949.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: sparse lexical and expansion model for first stage ranking. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo

Castells, Rosie Jones, and Tetsuya Sakai, editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2288–2292. ACM, 2021. doi: 10.1145/3404835.3463098. URL https://doi.org/10.1145/3404835.3463098.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1136. URL https://aclanthology.org/P19-1136/.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 12 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btl616. URL https://doi.org/10.1093/bioinformatics/btl616.

Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. Predicting completeness in knowledge bases. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 375–383, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346757. doi: 10.1145/3018661.3018739. URL https://doi.org/10.1145/3018661.3018739.

Chufan Gao, Xuan Wang, and Jimeng Sun. TTM-RE: Memory-augmented document-level relation extraction. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 443–458, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.26. URL https://aclanthology.org/2024.acl-long.26/.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 3816–3830, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295/.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997, 2023. doi: 10.48550/ARXIV.2312.10997. URL https://doi.org/10.48550/arXiv.2312.10997.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.366. URL https://aclanthology.org/2024.naacl-long.366/.

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. Self-verification improves few-shot clinical information extraction. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2023. URL https://openreview.net/forum?id=SBbJICrglS.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12216–12235. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.751. URL https://doi.org/10.18653/v1/2023.emnlp-main.751.

Shrestha Ghosh, Simon Razniewski, and Gerhard Weikum. Class cardinality comparison as a fermi problem. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben, editors, *Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 148–151. ACM, 2023. doi: 10.1145/3543873.3587334. URL https://doi.org/10.1145/3543873.3587334.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In Mohit Bansal and Aline Villavicencio, editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1049. URL https://aclanthology.org/K19-1049/.

John Giorgi, Gary Bader, and Bo Wang. A sequence-to-sequence approach for document-level relation extraction. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors, *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 10–25, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bionlp-1.2. URL https://aclanthology.org/2022.bionlp-1.2/.

Ankur Goswami, Akshata Bhat, Hadar Ohana, and Theodoros Rekatsinas. Unsupervised relation extraction from language models using constrained cloze completion. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1263–1276, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.113. URL https://aclanthology.org/2020.findings-emnlp.113/.

Ralph Grishman and Beth Sundheim. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. URL https://aclanthology.org/C96-1079/.

Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of* *the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, 2018. Association for Computational Linguistics. URL http://aclweb.org/anthology/N18-1065.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 267–274. ACM, 2009. doi: 10.1145/1571941.1571989. URL https://doi.org/10.1145/1571941.1571989.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. LongT5: Efficient text-to-text transformer for long sequences. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.55. URL https://aclanthology.org/2022.findings-naacl.55/.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. Generative prompt tuning for relation classification. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3170–3185, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.231. URL https://aclanthology.org/2022.findings-emnlp.231/.

Xiaoyu Han and Lei Wang. A novel document-level relation extraction method based on bert and entity information. *IEEE Access*, 8:96912–96919, 2020. doi: 10.1109/ACCESS.2020.2996642.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. More data, more relations, more context and more openness: A review and outlook for relation extraction. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China, December 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.aacl-main.75. URL https://aclanthology.org/2020.aacl-main.75/.

Shibo Hao, Bowen Tan, Kaiwen Tang, Bin Ni, Xiyan Shao, Hengzhe Zhang, Eric Xing, and Zhiting Hu. BertNet: Harvesting knowledge graphs with arbitrary relations from pretrained language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5000–5015, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.309. URL https://aclanthology.org/2023.findings-acl.309/.

Hartigan. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C, Applied statistics*, 28(1):100, 1979. ISSN 0035-9254. doi: 10.2307/2346830. URL https://tinyurl.sfx.mpg.de/x9rk.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. Efficient nearest neighbor language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.461. URL https://aclanthology.org/2021.emnlp-main.461/.

Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1076. URL https://aclanthology.org/D15-1076/.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and what's next. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1044. URL https://aclanthology.org/P17-1044/.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. Syntax for semantic role labeling, to be, or not to be. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2061–2071, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1192. URL https://aclanthology.org/P18-1192/.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Katrin Erk and Carlo Strapparava, editors, *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/S10-1006/.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In Regina Barzilay and Mark Johnson, editors, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1072/.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1055/.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. *Knowledge Graphs*. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers, 2021. ISBN 978-3-031-00790-3. doi: 10.2200/S01125ED1V01Y202109DSK022. URL https://doi.org/10.2200/S01125ED1V01Y202109DSK022.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.564. URL https://aclanthology.org/2021.emnlp-main.564/.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. *Zenodo*, 2020. doi: 10.5281/zenodo.1212303.

Andrew Hopkinson, Amit Gurdasani, Dave Palfrey, and Arpit Mittal. Demand-weighted completeness prediction for a knowledge base. In Srinivas Bangalore, Jennifer Chu-Carroll, and Yunyao Li, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages

200–207, New Orleans - Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-3025. URL https://aclanthology.org/N18-3025/.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. DEGREE: A data-efficient generation-based event extraction model. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.138. URL https://aclanthology.org/2022.naacl-main.138/.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015. URL http://arxiv.org/abs/1508.01991.

Scott B. Huffman. Learning information extraction patterns from examples. In Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, volume 1040 of *Lecture Notes in Computer Science*, pages 246–260. Springer, 1995. doi: 10.1007/3-540-60925-3_51. URL https://doi.org/10.1007/3-540-60925-3_51.

Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation extraction by end-to-end language generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.204. URL https://aclanthology.org/2021.findings-emnlp.204/.

Panagiotis G. Ipeirotis, Eugene Agichtein, Pranay Jain, and Luis Gravano. Towards a query optimizer for text-centric tasks. *ACM Transactions on Database Systems*, 32(4):21–es, 2007. doi: 10.1145/1292609.1292611.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74. URL https://aclanthology.org/2021.eacl-main.74/.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=jKN1pXi7b0.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6237–6250, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.418. URL https://aclanthology.org/2022.emnlp-main.418/.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions of Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418.

Basra Jehangir, Saravanan Radhakrishnan, and Rahul Agarwal. A survey on named entity recognition - datasets, tools, and methodologies. *Nat. Lang. Process. J.*, 3:100017, 2023. doi: 10.1016/J.NLP.2023.100017. URL https://doi.org/10.1016/j.nlp.2023.100017.

Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. In Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://aclanthology.org/P08-1030/.

Heng Ji, Ralph Grishman, Hoa T. Dang, Kira Griffitt, and Joe Ellis. Overview of the tac 2010 knowledge base population track. In *TAC*, 2010.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, and Cash Costello. Overview of TAC-KBP2017 13 languages entity discovery and linking. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*. NIST, 2017. URL https://tac.nist.gov/publications/2017/additional.papers/TAC2017.KBP_Entity_Discovery_and_Linking_overview.proceedings.pdf.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38, 2023. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. TEQUILA: temporal question answering over knowledge bases. In Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang, editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1807–1810. ACM, 2018. doi: 10.1145/3269206.3269247. URL https://doi.org/10.1145/3269206.3269247.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. Complex temporal question answering on knowledge graphs. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong, editors, *CIKM '21: The 30th ACM International Conference on*

Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, pages 792–802. ACM, 2021. doi: 10.1145/3459637.3482416. URL https://doi.org/10.1145/3459637.3482416.

Zhen Jia, Philipp Christmann, and Gerhard Weikum. Faithful temporal question answering over heterogeneous sources. In Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee, editors, *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 2052–2063. ACM, 2024. doi: 10.1145/3589334.3645547. URL https://doi.org/10.1145/3589334.3645547.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In Candace Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, Rochester, New York, April 2007. Association for Computational Linguistics. URL https://aclanthology.org/N07-1015/.

Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M. Kaplan, Timothy P. Hanratty, and Jiawei Han. Metapad: Meta pattern discovery from massive text corpora. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 877–886, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098105. URL https://doi.org/10.1145/3097983.3098105.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.479. URL https://aclanthology.org/2020.emnlp-main.479/.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020b. doi: 10.1162/tacl_a_00324. URL https://aclanthology.org/2020.tacl-1.28/.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 09 2021a. ISSN 2307-387X. doi: 10.1162/tacl_a_00407. URL https://doi.org/10.1162/tacl_a_00407.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021b. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57/.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. Instruct and extract: Instruction tuning for on-demand information extraction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.620. URL https://aclanthology.org/2023.emnlp-main.620/.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025. doi: 10.48550/ARXIV.2503.09516. URL https://doi.org/10.48550/arXiv.2503.09516.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. Lifelong pretraining: Continually adapting language models to emerging corpora. In Angela Fan, Suzana Ilic, Thomas Wolf, and Matthias Gallé, editors, *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 1–16, virtual+Dublin, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bigscience-1.1. URL https://aclanthology.org/2022.bigscience-1.1/.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5/.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. GenIE: Generative information extraction. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4643, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.342. URL https://aclanthology.org/2022.naacl-main.342/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *CoRR*, abs/2207.05221, 2022. doi: 10.48550/ARXIV.2207.05221. URL https://doi.org/10.48550/arXiv.2207.05221.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *CoRR*, abs/2307.10169, 2023. doi: 10.48550/ARXIV.2307.10169. URL https://doi.org/10.48550/arXiv.2307.10169.

Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/P04-3022/.

G. Kamradt. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR, 2023. URL https://proceedings.mlr.press/v202/kandpal23a.html.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. One thousand and one pairs: A "novel" challenge for long-context language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17048–17085, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.948. URL https://aclanthology.org/2024.emnlp-main.948/.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. Realtime QA: what's the answer right now? In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/9941624ef7f867a502732b5154d30cb7-Abstract-Datasets_and_Benchmarks.html.

Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.698. URL https://aclanthology.org/2020.acl-main.698/.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.284. URL https://aclanthology.org/2021.eacl-main.284/.

Mayank Kejriwal and Akarsh Nagaraj. Quantifying gender disparity in pre-modern english literature using natural language processing. *Journal of Data Science*, 22(1), 2024.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. A survey on recent advances in named entity recognition. *CoRR*, abs/2401.10825, 2024. doi: 10.48550/ARXIV.2401.10825. URL https://doi.org/10.48550/arXiv.2401.10825.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HklBjCEKvH.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.266. URL https://aclanthology.org/2022.naacl-main.266/.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401075. URL https://doi.org/10.1145/3397271.3401075.

Jun-Tae Kim and D.I. Moldovan. Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):713–724, 1995. doi: 10.1109/69.469825.

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varun Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Fables: Evaluating faithfulness and content selection in book-length summarization. In *Conference on Language Modeling*, 2024.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL https://aclanthology.org/Q18-1023/.

Tom Kocmi and Christian Federmann. Large language models are state-of-the-art evaluators of translation quality. In Mary Nurminen, Judith Brenner, Maarit Koponen, Sirkku Latomaa, Mikhail Mikhailov, Frederike Schierl, Tharindu Ranasinghe, Eva Vanmassenhove, Sergi Alvarez Vidal, Nora Aranberri, Mara Nunziatini, Carla Parra Escartín, Mikel Forcada, Maja Popovic, Carolina Scarton, and Helena Moniz, editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland, June 2023. European Association for Machine Translation. URL https://aclanthology.org/2023.eamt-1.19/.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In Anna Korhonen and Ivan Titov, editors, *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1050. URL https://aclanthology.org/K18-1050/.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3748–3761, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.306. URL https://aclanthology.org/2020.emnlp-main.306/.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. IMoJIE: Iterative memory-based joint open information extraction. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.521. URL https://aclanthology.org/2020.acl-main.521/.

Keshav Kolluru, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti, and Mausam. Alignment-augmented consistent translation for multilingual open information extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.179. URL https://aclanthology.org/2022.acl-long.179/.

Klaus Krippendorff. Computing krippendorff's alpha-reliability. In *Departmental Papers of the Annenberg School for Communication*, 2011. URL https://api.semanticscholar.org/CorpusID:59901023.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Alice Kwak, Clayton Morrison, Derek Bambauer, and Mihai Surdeanu. Classify first, and then extract: Prompt chaining technique for information extraction. In Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Cătălina Goanță, Daniel Preoțiuc-Pietro, and Gerasimos Spanakis, editors, *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 303–317, Miami, FL, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nllp-1.25. URL https://aclanthology.org/2024.nllp-1.25/.

Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the web. *ACM Trans. Inf. Syst.*, 19(3):242–262, 2001. doi: 10.1145/502115.502117. URL https://doi.org/10.1145/502115.502117.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. A survey on complex knowledge base question answering: Methods, challenges and solutions. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4483–4491. ijcai.org, 2021. doi: 10.24963/IJCAI.2021/611. URL https://doi.org/10.24963/ijcai.2021/611.

Nghia T. Le and Alan Ritter. Are language models robust coreference resolvers? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MmBQSNHKUl.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In Sameer Pradhan, editor, *Proceedings of the Fifteenth Conference on Computational Natural*

Language Learning: Shared Task, pages 28–34, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/W11-1902/.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1018. URL https://aclanthology.org/D17-1018/.

Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2108. URL https://aclanthology.org/N18-2108/.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press, 2012. ISBN 9781577355601.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. The inductive bias of in-context learning: Rethinking pretraining example design. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=lnEaqbTJIRz.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068, 05 2016. ISSN 1758-0463. doi: 10.1093/database/baw068. URL https://doi.org/10.1093/database/baw068.

Junpeng Li, Zixia Jia, and Zilong Zheng. Semi-automatic data enhancement for document-level relation extraction with distant supervision from large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5495–5505, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.334. URL https://aclanthology.org/2023.emnlp-main.334/.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, Tat-Seng Chua, and Yang Deng. Knowledge boundary of large language models: A survey. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5131–5157, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.256. URL https://aclanthology.org/2025.acl-long.256/.

Tianyi Li, Wenyu Huang, Nikos Papasarantopoulos, Pavlos Vougiouklis, and Jeff Z. Pan. Task-specific pre-training and prompt decomposition for knowledge graph population with language models. In Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski, editors, *Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022 co-located with the 21st International Semantic Web Conference (ISWC2022), Virtual Event, Hanghzou, China, October 2022*, volume 3274 of *CEUR Workshop Proceedings*, pages 35–45. CEUR-WS.org, 2022a. URL https://ceur-ws.org/Vol-3274/paper3.pdf.

Wenbiao Li, Wang Ziyang, and Yunfang Wu. A unified neural network model for readability assessment with feature projection and length-balanced loss. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7446–7457, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.504. URL https://aclanthology.org/2022.emnlp-main.504/.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Xiaoyan Li and W. Bruce Croft. Time-based language models. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003*, pages 469–475. ACM, 2003. doi: 10.1145/956863.956951. URL https://doi.org/10.1145/956863.956951.

Xinze Li, Yixin Cao, Yubo Ma, and Aixin Sun. Long context vs. RAG for llms: An evaluation and revisits. *CoRR*, abs/2501.01880, 2025b. doi: 10.48550/ARXIV.2501.01880. URL https://doi.org/10.48550/arXiv.2501.01880.

Yinheng Li. A practical survey on zero-shot prompt design for in-context learning. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 641–647, Varna, Bulgaria, September 2023. INCOMA Ltd., Shoumen, Bulgaria. URL https://aclanthology.org/2023.ranlp-1.69/.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.66. URL https://aclanthology.org/2024.emnlp-industry.66/.

Shasha Liao and Ralph Grishman. Using document level cross-event inference to improve event extraction. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre, editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1081/.

Jimmy Lin and Xueguang Ma. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *CoRR*, abs/2106.14807, 2021. URL https://arxiv.org/abs/2106.14807.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2356–2362. ACM, 2021. doi: 10.1145/3404835.3463238. URL https://doi.org/10.1145/3404835.3463238.

Thomas Lin, Patrick Pantel, Michael Gamon, Anitha Kannan, and Ariel Fuxman. Active objects: actions for entity-centric search. In Alain Mille, Fabien Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 589–598. ACM, 2012. doi: 10.1145/2187836.2187916. URL https://doi.org/10.1145/2187836.2187916.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1200. URL https://aclanthology.org/P16-1200/.

Xiao Ling, Sameer Singh, and Daniel S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015. doi: 10.1162/tacl_a_00141. URL https://aclanthology.org/Q15-1023/.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Narayanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases*, pages 225–239, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić, editors, *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10/.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. Event extraction as machine reading comprehension. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.128. URL https://aclanthology.org/2020.emnlp-main.128/.

Jian Liu, Yufeng Chen, and Jinan Xu. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.214. URL https://aclanthology.org/2021.emnlp-main.214/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

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. *ACM Comput. Surv.*, 55(9), January 2023a. ISSN 0360-0300. doi: 10.1145/3560815. URL https://doi.org/10.1145/3560815.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8/.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Samuel Louvan and Bernardo Magnini. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.42. URL https://aclanthology.org/2020.coling-main.42/.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556/.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 2795–2806, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.217. URL https://aclanthology.org/2021.acl-long.217/.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland, May

2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.395. URL https://aclanthology.org/2022.acl-long.395/.

Michael Luggen, Djellel Difallah, Cristina Sarasua, Gianluca Demartini, and Philippe Cudré-Mauroux. Non-parametric class completeness estimators for collaborative knowledge graphs—the case of wikidata. In *The Semantic Web – ISWC 2019*, pages 453–469, Cham, 2019. Springer International Publishing.

Man Luo, Kazuma Hashimoto, Semih Yavuz, Zhiwei Liu, Chitta Baral, and Yingbo Zhou. Choose your QA model wisely: A systematic study of generative and extractive readers for question answering. In Rajarshi Das, Patrick Lewis, Sewon Min, June Thai, and Manzil Zaheer, editors, *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 7–22, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.spanlp-1.2. URL https://aclanthology.org/2022.spanlp-1.2/.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. In-context learning with retrieved demonstrations for language models: A survey. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=NQPo8ZhQPa.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A. Smith. Time waits for no one! analysis and challenges of temporal misalignment. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5944–5958, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.435. URL https://aclanthology.org/2022.naacl-main.435/.

Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, and Dimitris Samaras. Docunet: Document image unwarping via a stacked u-net. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4709, 2018. doi: 10.1109/CVPR.2018.00494.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. Template-free prompt tuning for few-shot NER. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.420. URL https://aclanthology.org/2022.naacl-main.420/.

Xilai Ma, Jing Li, and Min Zhang. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2334–2352, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.153. URL https://aclanthology.org/2023.findings-emnlp.153/.

Youmi Ma, An Wang, and Naoaki Okazaki. DREEAM: Guiding attention with evidence for improving document-level relation extraction. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1971–1983, Dubrovnik, Croatia, May 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.145. URL https://aclanthology.org/2023.eacl-main.145/.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. Controllable text simplification with explicit paraphrasing. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.277. URL https://aclanthology.org/2021.naacl-main.277/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.546. URL https://doi.org/10.18653/v1/2023.acl-long.546.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In Kalina Bontcheva and Jingbo Zhu, editors, *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-5010. URL https://aclanthology.org/P14-5010/.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008. ISBN 978-0-521-86571-5. doi: 10.1017/CBO9780511809071. URL https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf.

Lev Manovich. *Cultural analytics*. Mit Press, 2020.

Jose L. Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. Information extraction meets the semantic web: A survey. *Semantic Web Journal*, 2020. doi: 10.3233/SW-180333.

José-Lázaro Martínez-Rodríguez, Ivan López-Arévalo, and Ana B. Ríos-Alvarado. Openie-based approach for knowledge graph construction from text. *Expert Syst. Appl.*, 113:339–355, 2018. doi: 10.1016/J.ESWA.2018.07.017. URL https://doi.org/10.1016/j.eswa.2018.07.017.

Mausam. Open information extraction systems and downstream applications. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4074–4077. IJCAI/AAAI Press, 2016. URL http://www.ijcai.org/Abstract/16/604.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In Jun'ichi Tsujii, James Henderson, and Marius Paşca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/D12-1048/.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N. Ioannidis, Adesoji Adeshina, Phillip Ryan Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. Tempoqr: Temporal question reasoning over knowledge graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 5825–5833. AAAI Press, 2022. doi: 10.1609/AAAI.V36I5.20526. URL https://doi.org/10.1609/aaai.v36i5.20526.

Olena Medelyan, David N. Milne, Catherine Legg, and Ian H. Witten. Mining meaning from wikipedia. *Int. J. Hum. Comput. Stud.*, 67(9):716–754, 2009. doi: 10.1016/J.IJHCS.2009.05.004. URL https://doi.org/10.1016/j.ijhcs.2009.05.004.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. Rethinking search: making domain experts out of dilettantes. *SIGIR Forum*, 55(1):13:1–13:27, 2021. doi: 10.1145/3476415.3476428. URL https://doi.org/10.1145/3476415.3476428.

Gaspard Michel, Elena V. Epure, Romain Hennequin, and Christophe Cerisara. Evaluating LLMs for quotation attribution in literary texts: A case study of LLaMa3. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 742–755, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

ISBN 979-8-89176-190-2. doi: 10.18653/v1/2025.naacl-short.62. URL https://aclanthology.org/2025.naacl-short.62/.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a. URL http://arxiv.org/abs/1301.3781.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013b. URL https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1095/.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.201. URL https://aclanthology.org/2022.naacl-main.201/.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Shervin Minaee, Tomás Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *CoRR*, abs/2402.06196, 2024. doi: 10.48550/ARXIV.2402.06196. URL https://doi.org/10.48550/arXiv.2402.06196.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In Keh-Yih Su, Jian Su, Janyce Wiebe, and Haizhou Li, editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://aclanthology.org/P09-1113/.

Paramita Mirza, Simon Razniewski, Fariz Darari, and Gerhard Weikum. Enriching knowledge bases with counting quantifiers. In Denny Vrandecic, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, pages 179–197. Springer, 2018. doi: 10.1007/978-3-030-00671-6_11. URL https://doi.org/10.1007/978-3-030-00671-6_11.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka, Partha P. Talukdar, Bo Yang, J. Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry T. Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018. doi: 10.1145/3191513.

Makoto Miwa and Mohit Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1105. URL https://aclanthology.org/P16-1105/.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2: 231–244, 2014. doi: 10.1162/tacl_a_00179. URL https://aclanthology.org/Q14-1019/.

Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159, 06 2008. ISSN 0891-2017. doi: 10.1162/coli.2008.34.2.145. URL https://doi.org/10.1162/coli.2008.34.2.145.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26, 2007. URL https://api.semanticscholar.org/CorpusID:8310135.

Marc Najork. Generative information retrieval. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete, editors, *Proceedings of the 46th*

International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, page 1. ACM, 2023. doi: 10.1145/3539618.3591871. URL https://doi.org/10.1145/3539618.3591871.

Ndapandula Nakashole and Tom M. Mitchell. Language-aware truth assessment of fact candidates. In Kristina Toutanova and Hua Wu, editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1019, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1095. URL https://aclanthology.org/P14-1095/.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. PATTY: A taxonomy of relational patterns with semantic types. In Jun'ichi Tsujii, James Henderson, and Marius Paşca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/D12-1104/.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. Reasoning with latent structure refinement for document-level relation extraction. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.141. URL https://aclanthology.org/2020.acl-main.141/.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. *CoRR*, abs/2201.10005, 2022.

Benjamin Newman, Prafulla Kumar Choubey, and Nazneen Rajani. P-adapters: Robustly extracting factual information from language models with diverse prompts. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=DhzIU48OcZh.

Vincent Ng. Machine learning for entity coreference resolution: A retrospective look at two decades of research. In Satinder Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4877–4884. AAAI Press, 2017. doi: 10.1609/AAAI.V31I1.11149. URL https://doi.org/10.1609/aaai.v31i1.11149.

Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from wikipedia using subtree mining. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1414–1420. AAAI Press, 2007. URL http://www.aaai.org/Library/AAAI/2007/aaai07-224.php.

Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2060. URL https://aclanthology.org/P15-2060/.

Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi. Joint learning of local and global features for entity linking via neural networks. In Yuji Matsumoto and Rashmi Prasad, editors, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2310–2320, Osaka, Japan, December 2016a. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1218/.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne, editors, *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016b. URL https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33, 2016. doi: 10.1109/JPROC.2015.2483592. URL https://doi.org/10.1109/JPROC.2015.2483592.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1326/.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. TORQUE: A reading comprehension dataset of temporal ordering questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.88. URL https://aclanthology.org/2020.emnlp-main.88/.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019. URL http://arxiv.org/abs/1901.04085.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings* *of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.63. URL https://aclanthology.org/2020.findings-emnlp.63/.

Douglas W. Oard, Gina-Anne Levow, and Clara I. Cabezas. Clef experiments at maryland: Statistical stemming and backoff translation. In Carol Peters, editor, *Cross-Language Information Retrieval and Evaluation*, pages 176–187, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44645-3.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.115. URL https://aclanthology.org/2022.findings-naacl.115/.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023a. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

OpenAI. Function calling and other API updates (longer context). https://openai.com/blog/function-calling-and-other-api-updates, 2023b. Accessed on: 21 September 2025.

Laurel J. Orr, Megan Leszczynski, Neel Guha, Sen Wu, Simran Arora, Xiao Ling, and Christopher Ré. Bootleg: Chasing the tail with self-supervised named entity disambiguation. In *11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15, 2021, Online Proceedings*. www.cidrdb.org, 2021. URL http://cidrdb.org/cidr2021/papers/cidr2021_paper13.pdf.

Liu Pai, Wenyang Gao, Wenjie Dong, Lin Ai, Ziwei Gong, Songfang Huang, Li Zongsheng, Ehsan Hoque, Julia Hirschberg, and Yue Zhang. A survey on open information extraction from rule-based model to large language model. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9586–9608, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.560. URL https://aclanthology.org/2024.findings-emnlp.560/.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005. doi: 10.1162/0891201053630264. URL https://aclanthology.org/J05-1004/.

Martha Palmer, Ivan Titov, and Shumin Wu. Semantic role labeling. In Jimmy Lin and Katrin Erk, editors, *NAACL HLT 2013 Tutorial Abstracts*, pages 10–12, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-4004/.

Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. Large language models and knowledge graphs: Opportunities and challenges. *TGDK*, 1(1):2:1–2:38, 2023. doi: 10.4230/TGDK.1.1.2. URL https://doi.org/10.4230/TGDK.1.1.2.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.*, 36(7):3580–3599, 2024. doi: 10.1109/TKDE.2024.3352100. URL https://doi.org/10.1109/TKDE.2024.3352100.

Rrubaa Panchendrarajan and Aravindh Amaresan. Bidirectional LSTM-CRF for named entity recognition. In Stephen Politzer-Ahles, Yu-Yin Hsu, Chu-Ren Huang, and Yao Yao, editors, *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, 1–3 December 2018. Association for Computational Linguistics. URL https://aclanthology.org/Y18-1061/.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. QuALITY: Question answering with long input texts, yes! In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.391. URL https://aclanthology.org/2022.naacl-main.391/.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=US-TP-xnXI.

Kevin Pei, Ishan Jindal, Kevin Chen-Chuan Chang, ChengXiang Zhai, and Yunyao Li. When to use what: An in-depth comparative empirical analysis of OpenIE systems for downstream applications. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 929–949, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.53. URL https://aclanthology.org/2023.acl-long.53/.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017. doi: 10.1162/tacl_a_00049. URL https://aclanthology.org/Q17-1008/.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202/.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250/.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL https://aclanthology.org/2021.naacl-main.200/.

Andrew Piper and Sunyam Bagga. Using large language models for understanding narrative discourse. In Yash Kumar Lal, Elizabeth Clark, Mohit Iyyer, Snigdha Chaturvedi, Anneliese Brei, Faeze Brahman, and Khyathi Raghavi Chandu, editors, *Proceedings of the 6th Workshop on Narrative Understanding*, pages 37–46, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.wnu-1.4. URL https://aclanthology.org/2024.wnu-1.4/.

Andrew Piper, Richard Jean So, and David Bamman. Narrative theory for computational narrative understanding. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 298–311, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.26. URL https://aclanthology.org/2021.emnlp-main.26/.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. Exploiting the syntax-model consistency for neural relation extraction. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8021–8032, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.715. URL https://aclanthology.org/2020.acl-main.715/.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In Sameer Pradhan, Alessandro Moschitti, and Nianwen Xue, editors, *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/W12-4501/.

Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.410. URL https://aclanthology.org/2021.naacl-main.410/.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S. Yu. Large language models meet NLP: A survey. *CoRR*, abs/2405.12819, 2024. doi: 10.48550/ARXIV.2405.12819. URL https://doi.org/10.48550/arXiv.2405.12819.

Chris Quirk and Hoifung Poon. Distant supervision for relation extraction beyond the sentence boundary. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-1110/.

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In Hang Li and Lluís Màrquez, editors, *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Cambridge, MA, October 2010. Association for Computational Linguistics. URL https://aclanthology.org/D10-1048/.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings* *of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023. doi: 10.1162/tacl_a_00605. URL https://aclanthology.org/2023.tacl-1.75/.

Delip Rao, Paul McNamee, and Mark Dredze. Entity linking: Finding extracted entities in a knowledge base. In Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 93–115. Springer, 2013. doi: 10.1007/978-3-642-28569-1_5. URL https://doi.org/10.1007/978-3-642-28569-1_5.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1317. URL https://aclanthology.org/D17-1317/.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to Wikipedia. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1138/.

Simon Razniewski, Vevake Balaraman, and Werner Nutt. Doctoral advisor or medical condition: Towards entity-specific rankings of knowledge base properties. In Gao Cong, Wen-Chih Peng, Wei Emma Zhang, Chengliang Li, and Aixin Sun, editors, *Advanced Data Mining and Applications - 13th International Conference, ADMA 2017, Singapore, November 5-6, 2017, Proceedings*, volume 10604 of *Lecture Notes in Computer Science*, pages 526–540. Springer, 2017. doi: 10.1007/978-3-319-69179-4_37. URL https://doi.org/10.1007/978-3-319-69179-4_37.

Simon Razniewski, Nitisha Jain, Paramita Mirza, and Gerhard Weikum. Coverage of information extraction from sentences and paragraphs. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5771–5776, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1583. URL https://aclanthology.org/D19-1583/.

Simon Razniewski, Andrew Yates, Nora Kassner, and Gerhard Weikum. Language models as or for knowledge bases. In Mehwish Alam, Davide Buscaldi, Michael Cochez, Diego Reforgiato

Recupero, and Harald Sack, editors, *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, online, October 25, 2021*, volume 3034 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021. URL https://ceur-ws.org/Vol-3034/paper2.pdf.

Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, and Jeff Z. Pan. 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. In *CEUR Workshop Proceedings*, volume 3577. CEUR-WS.org, 2023. URL https://ceur-ws.org/Vol-3577.

Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jeff Z. Pan, Tuan-Phong Nguyen, and Bohui Zhang. Joint proceedings of the 2nd workshop on knowledge base construction from pre-trained language models (KBC-LM 2024) and the 3rd challenge on language models for knowledge base construction (LM-KBC 2024) co-located with the 23nd international semantic web conference (ISWC 2024), baltimore, usa, november 12, 2024. In *CEUR Workshop Proceedings*, volume 3853. CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3853.

Simon Razniewski, Jan-Christoph Kalo, Duygu Islakoğlu, Tuan-Phong Nguyen, and Bohui Zhang. Joint proceedings of the 3rd workshop on knowledge base construction from pre-trained language models and the 4th challenge on language models for knowledge base construction (kbc-lm+lm-kbc 2025) co-located with the 24th international semantic web conference (iswc 2025), nara, japan, november 2, 2025. In *CEUR Workshop Proceedings*, volume 4041. CEUR-WS.org, 2025. URL https://ceur-ws.org/Vol-4041.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530, 2024. doi: 10.48550/ARXIV.2403.05530. URL https://doi.org/10.48550/arXiv.2403.05530.

Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Pro-*

cessing, pages 338–348, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1035. URL https://aclanthology.org/D17-1035/.

Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–306, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.17. URL https://aclanthology.org/2023.acl-long.17/.

Philip Resnik, Douglas Oard, and Gina Levow. Improved cross-language retrieval using back-off translation. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001. URL https://aclanthology.org/H01-1033/.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer, 2010. doi: 10.1007/978-3-642-15939-8_10. URL https://doi.org/10.1007/978-3-642-15939-8_10.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Dwaipayan Roy, Sumit Bhatia, and Prateek Jain. A topic-aligned multilingual corpus of Wikipedia articles for studying information asymmetry in low resource languages. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2373–2380, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.289/.

Ohad Rubin and Jonathan Berant. Retrieval-pretrained transformer: Long-range language modeling with self-retrieval. *Transactions of the Association for Computational Linguistics*, 12:1197–1213, 2024. doi: 10.1162/tacl_a_00693. URL https://aclanthology.org/2024.tacl-1.66/.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.191. URL https://aclanthology.org/2022.naacl-main.191/.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *CoRR*, abs/2402.07927, 2024. doi: 10.48550/ARXIV.2402.07927. URL https://doi.org/10.48550/arXiv.2402.07927.

Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium*, 6(12):e26752, 2008. doi: 10.35111/77ba-9x74.

Ville Satopaa, Jeannie R. Albrecht, David E. Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *31st IEEE International Conference on Distributed Computing Systems Workshops (ICDCS 2011 Workshops), 20-24 June 2011, Minneapolis, Minnesota, USA*, pages 166–171. IEEE Computer Society, 2011. doi: 10.1109/ICDCSW.2011.20. URL https://doi.org/10.1109/ICDCSW.2011.20.

Uma Sawant and Soumen Chakrabarti. Learning joint query interpretation and response ranking. In Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo Baeza-Yates, and Sue B. Moon, editors, *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1099–1110. International World Wide Web Conferences Steering Committee / ACM, 2013. doi: 10.1145/2488388.2488484. URL https://doi.org/10.1145/2488388.2488484.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 6663–6676, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.520. URL https://aclanthology.org/2021.acl-long.520/.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online, April 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.20. URL https://aclanthology.org/2021.eacl-main.20/.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online, June 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.185. URL https://aclanthology.org/2021.naacl-main.185/.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer, 2018. doi: 10.1007/978-3-319-93417-4_38. URL https://doi.org/10.1007/978-3-319-93417-4_38.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. Analysing errors of open information extraction systems. In Emily Bender, Hal Daumé III, Allyson Ettinger, and Sudha Rao, editors, *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 11–18, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5402. URL https://aclanthology.org/W17-5402/.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson C. Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Miserlis Hoyle, and Philip Resnik. The prompt report: A systematic survey of prompting techniques. *CoRR*, abs/2406.06608, 2024. doi: 10.48550/ARXIV.2406.06608. URL https://doi.org/10.48550/arXiv.2406.06608.

Özge Sevgili, Artem Shelmanov, Mikhail Y. Arkhipov, Alexander Panchenko, and Chris Biemann. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570, 2022. doi: 10.3233/SW-222986. URL https://doi.org/10.3233/SW-222986.

Jianhao Shen, Chenguang Wang, Ye Yuan, Jiawei Han, Heng Ji, Koushik Sen, Ming Zhang, and Dawn Song. PALT: Parameter-lite transfer of language models for knowledge graph completion. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3833–3847, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.281. URL https://aclanthology.org/2022.findings-emnlp.281/.

Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.*, 27(2):443–460, 2015. doi: 10.1109/TKDE.2014.2327028. URL https://doi.org/10.1109/TKDE.2014.2327028.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Wen tau Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=LXVswInHOo.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico, June 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.463. URL https://aclanthology.org/2024.naacl-long.463/.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.346. URL https://aclanthology.org/2020.emnlp-main.346/.

Mariana O Silva, Luiza de Melo-Gomes, and Mirella M Moro. Gender representation in literature: Analysis of characters' physical descriptions. In *Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pages 17–24. SBC, 2023.

Amit Singhal. Introducing the knowledge graph: things, not strings, 2012. URL https://www.blog.google/products/search/introducing-knowledge-graph-things-not/. Accessed on: 03 December 2025.

Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski. Proceedings of the semantic web challenge on knowledge base construction from pre-trained language models 2022 co-located with the 21st international semantic web conference (iswc2022), virtual event, hanghzou, china, october 2022. In *CEUR Workshop Proceedings*, volume 3274. CEUR-WS.org, 2022a. URL https://ceur-ws.org/Vol-3274.

Sneha Singhania, Simon Razniewski, and Gerhard Weikum. Predicting document coverage for relation extraction. *Transactions of the Association for Computational Linguistics*, 10:207–223, 2022b. doi: 10.1162/tacl_a_00456. URL https://aclanthology.org/2022.tacl-1.12/.

Sneha Singhania, Simon Razniewski, and Gerhard Weikum. Extracting multi-valued relations from language models. In Burcu Can, Maximilian Mozes, Samuel Cahyawijaya, Naomi Saphra, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Chen Zhao, Isabelle Augenstein, Anna Rogers, Kyunghyun Cho, Edward Grefenstette, and Lena Voita, editors, *Proceedings of the 8th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2023, Toronto, Canada, July 13, 2023*, pages 139–

154. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.REPL4NLP-1.12. URL https://doi.org/10.18653/v1/2023.repl4nlp-1.12.

Sneha Singhania, Simon Razniewski, and Gerhard Weikum. Extracting multi-valued relations from language models. In Burcu Can, Maximilian Mozes, Samuel Cahyawijaya, Naomi Saphra, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Chen Zhao, Isabelle Augenstein, Anna Rogers, Kyunghyun Cho, Edward Grefenstette, and Lena Voita, editors, *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 139–154, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.repl4nlp-1.12. URL https://aclanthology.org/2023.repl4nlp-1.12/.

Sneha Singhania, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. Neon: News entity-interaction extraction for enhanced question answering. In *Proceedings of the 1st Workshop on Robust Information Retrieval, RobustIR at SIGIR, Padua, Italy*, 2025a. URL https://doi.org/10.48550/arXiv.2411.12449.

Sneha Singhania, Simon Razniewski, and Gerhard Weikum. Recall them all: Long list generation from long novels. In *Proceedings of the 1st Workshop on Natural Language Processing and Language Models for Digital Humanities, CLARIN Workshop at RANLP, Varna, Bulgaria*. Association for Computational Linguistics, 2025b. URL https://arxiv.org/abs/2405.02732.

Sneha Singhania, Simon Razniewski, and Gerhard Weikum. L3x: Long object list extraction from long documents. In *ACM International Conference on Information and Knowledge Management*, 2025c. URL https://d5demos.mpi-inf.mpg.de/l3x.

Alisa Smirnova and Philippe Cudré-Mauroux. Relation extraction using distant supervision: A survey. *ACM Comput. Surv.*, 51(5):106:1–106:35, 2019. doi: 10.1145/3241741. URL https://doi.org/10.1145/3241741.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 926–934, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/b337e84de8752b27eda3a12363109e80-Abstract.html.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy G. Lehnert. CRYSTAL: inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, pages 1314–1321. Morgan Kaufmann, 1995. URL http://ijcai.org/Proceedings/95-2/Papers/038.pdf.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001. doi: 10.1162/089120101753342653. URL https://aclanthology.org/J01-4004/.

Dominik Stammbach, Maria Antoniak, and Elliott Ash. Heroes, villains, and victims, and GPT-3: Automated extraction of character roles without training data. In Elizabeth Clark, Faeze Brahman, and Mohit Iyyer, editors, *Proceedings of the 4th Workshop of Narrative Understanding (WNU2022)*, pages 47–56, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.wnu-1.6. URL https://aclanthology.org/2022.wnu-1.6/.

Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1252. URL https://aclanthology.org/D16-1252/.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1081. URL https://aclanthology.org/N18-1081/.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706. ACM, 2007. doi: 10.1145/1242572.1242667. URL https://doi.org/10.1145/1242572.1242667.

Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. Sofie: a self-organizing framework for information extraction. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 631–640, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584874. doi: 10.1145/1526709.1526794. URL https://doi.org/10.1145/1526709.1526794.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1242. URL https://aclanthology.org/D19-1242/.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (llm)? A.K.A. will llms replace knowledge graphs? *CoRR*, abs/2308.10168, 2023. doi: 10.48550/ARXIV.2308.10168. URL https://doi.org/10.48550/arXiv.2308.10168.

Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs? In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 311–325, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.18. URL https://aclanthology.org/2024.naacl-long.18/.

Simeng Sun, Katherine Thai, and Mohit Iyyer. ChapterBreak: A challenge dataset for long-range language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3704–3714, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.271. URL https://aclanthology.org/2022.naacl-main.271/.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137, 2021. URL https://arxiv.org/abs/2107.02137.

Beth M. Sundheim. The Message Understanding Conferences. In *TIPSTER TEXT PROGRAM: PHASE I: Proceedings of a Workshop held at Fredricksburg, Virginia, September 19-23, 1993*, pages 5–5, Fredericksburg, Virginia, USA, September 1993. Association for Computational Linguistics. doi: 10.3115/1119149.1119153. URL https://aclanthology.org/X93-1003/.

Mihai Surdeanu. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Proceedings of the 2013 Text Analysis Conference, TAC 2013*. NIST, 2013. URL https://tac.nist.gov/publications/2013/additional.papers/KBP2013_English_and_Temporal_Slot_Filling_overview.TAC2013.proceedings.pdf.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In Jun'ichi Tsujii, James Henderson, and Marius Paşca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/D12-1042/.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In Haizhou Li, Chin-Yew Lin, Miles Osborne, Gary Geunbae Lee, and Jong C. Park, editors, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–729, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/P12-1076/.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL https://aclanthology.org/N18-1059/.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758, 2020. doi: 10.1162/tacl_a_00342. URL https://aclanthology.org/2020.tacl-1.48/.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.407. URL https://aclanthology.org/2021.emnlp-main.407/.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. Evaluating the factual consistency of large language models through news summarization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.322. URL https://aclanthology.org/2023.findings-acl.322/.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. Revisiting DocRED - addressing the false negative problem in relation extraction. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8472–8487, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.580. URL https://aclanthology.org/2022.emnlp-main.580/.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. HIN: hierarchical inference network for document-level relation extraction. In Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan, editors, *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part I*, volume 12084 of *Lecture Notes in Computer Science*, pages 197–209. Springer, 2020. doi: 10.1007/978-3-030-47426-3_16. URL https://doi.org/10.1007/978-3-030-47426-3_16.

Katherine Thai, Yapei Chang, Kalpesh Krishna, and Mohit Iyyer. RELiC: Retrieving evidence for literary claims. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7500–7518, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.517. URL https://aclanthology.org/2022.acl-long.517/.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074/.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002. URL https://aclanthology.org/W02-2024/.

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*, pages 142–147, 2003. URL https://aclanthology.org/W03-0419/.

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. On generalization in coreference resolution. In Maciej Ogrodniczuk, Sameer Pradhan, Massimo Poesio, Yulia Grishina, and Vincent Ng, editors, *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 111–120, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.crac-1.12. URL https://aclanthology.org/2021.crac-1.12/.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. Gerbil: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 1133–1143, Republic and Canton of Geneva, CHE, 2015. International World

Wide Web Conferences Steering Committee. ISBN 9781450334693. doi: 10.1145/2736277.2741626. URL https://doi.org/10.1145/2736277.2741626.

Blerta Veseli, Simon Razniewski, Jan-Christoph Kalo, and Gerhard Weikum. Evaluating the knowledge base completion potential of GPT. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6432–6443, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.426. URL https://aclanthology.org/2023.findings-emnlp.426/.

Blerta Veseli, Sneha Singhania, Simon Razniewski, and Gerhard Weikum. Evaluating language models for knowledge base completion. In Catia Pesquita, Ernesto Jiménez-Ruiz, Jamie P. McCusker, Daniel Faria, Mauro Dragoni, Anastasia Dimou, Raphaël Troncy, and Sven Hertling, editors, *The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*, volume 13870 of *Lecture Notes in Computer Science*, pages 227–243. Springer, 2023b. doi: 10.1007/978-3-031-33455-9_14. URL https://doi.org/10.1007/978-3-031-33455-9_14.

Paul M. B. Vitányi and Rudi Cilibrasi. Normalized web distance and word similarity. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, pages 293–314. Chapman and Hall/CRC, 2010. doi: 10.1201/9781420085938-C13. URL http://www.crcnetbase.com/doi/abs/10.1201/9781420085938-c13.

Ellen M. Voorhees. Query expansion using lexical-semantic relations. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 61–69. ACM/Springer, 1994. doi: 10.1007/978-1-4471-2099-5_7. URL https://doi.org/10.1007/978-1-4471-2099-5_7.

Denny Vrandecic. Wikidata: a new platform for collaborative data collection. In Alain Mille, Fabien Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 1063–1064. ACM, 2012. doi: 10.1145/2187980.2188242. URL https://doi.org/10.1145/2187980.2188242.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of ACM*, 57(10):78–85, 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1585. URL https://aclanthology.org/D19-1585/.

Somin Wadhwa, Silvio Amir, and Byron Wallace. Revisiting relation extraction in the era of large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.868. URL https://aclanthology.org/2023.acl-long.868/.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://aclanthology.org/D19-1221/.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. Tnt-llm: Text mining at scale with large language models. In Ricardo Baeza-Yates and Francesco Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 5836–5847. ACM, 2024. doi: 10.1145/3637528.3671647. URL https://doi.org/10.1145/3637528.3671647.

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. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.214. URL https://aclanthology.org/2023.emnlp-main.214/.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *CoRR*, abs/2310.07521, 2023a. doi: 10.48550/ARXIV.2310.07521. URL https://doi.org/10.48550/arXiv.2310.07521.

Difeng Wang, Wei Hu, Ermei Cao, and Weijian Sun. Global-to-local neural networks for document-level relation extraction. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3711–3721, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.303. URL https://aclanthology.org/2020.emnlp-main.303/.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. GPT-NER: Named entity recognition via large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational*

Linguistics: NAACL 2025, pages 4257–4275, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.239. URL https://aclanthology.org/2025.findings-naacl.239/.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. Instructuie: Multi-task instruction tuning for unified information extraction. *CoRR*, abs/2304.08085, 2023b. doi: 10.48550/ARXIV.2304.08085. URL https://doi.org/10.48550/arXiv.2304.08085.

Xiaolan Wang, Xin L. Dong, Yang Li, and Alexandra Meliou. MIDAS: Finding the right web sources to fill knowledge gaps. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 578–589, 2019. doi: 10.1109/ICDE.2019.00058.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer2: Open-source dataset for training top-performing reward models. *CoRR*, abs/2406.08673, 2024. doi: 10.48550/ARXIV.2406.08673. URL https://doi.org/10.48550/arXiv.2406.08673.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. Redpajama: an open dataset for training large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/d34497330b1fd6530f7afd86d0df9f76-Abstract-Datasets_and_Benchmarks_Track.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Gerhard Weikum, Gjergji Kasneci, Maya Ramanath, and Fabian M. Suchanek. Database and information-retrieval methods for knowledge discovery. *Commun. ACM*, 52:56–64, 2009. doi: 10.1145/1498765.1498784.

Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Found. Trends Databases*, 10(2-4): 108–490, 2021. doi: 10.1561/1900000064. URL https://doi.org/10.1561/1900000064.

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Comput. Surv.*, 55(8), December 2022. ISSN 0360-0300. doi: 10.1145/3547138. URL https://doi.org/10.1145/3547138.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.167. URL https://aclanthology.org/2023.findings-emnlp.167/.

Matthew Wilkens, Elizabeth F Evans, Sandeep Soni, David Bamman, and Andrew Piper. Small worlds: Measuring the mobility of characters in english-language fiction. *Journal of Computational Literary Studies*, 3(1), 2024.

Fei Wu and Daniel S. Weld. Open information extraction using Wikipedia. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre, editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1013/.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.519. URL https://aclanthology.org/2020.emnlp-main.519/.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. How easily do irrelevant inputs skew the responses of large language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=S7NVVfuRv8.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=TrjbxzRcnf-.

Johnathan Xie, Annie S Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. Calibrating language models with adaptive temperature scaling. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18128–18138, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1007. URL https://aclanthology.org/2024.emnlp-main.1007/.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Empirical study of zero-shot NER with ChatGPT. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.493. URL https://aclanthology.org/2023.emnlp-main.493/.

Yiqing Xie, Jiaming Shen, Sha Li, Yuning Mao, and Jiawei Han. Eider: Empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 257–268, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.23. URL https://aclanthology.org/2022.findings-acl.23/.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10452–10470, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.563. URL https://aclanthology.org/2024.acl-long.563/.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BJlzm64tDH.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(16), pages 14149–14157, 2021a. URL https://ojs.aaai.org/index.php/AAAI/article/view/17665.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: a survey. *Frontiers Comput. Sci.*, 18(6):186357, 2024a. doi: 10.1007/S11704-024-40555-Y. URL https://doi.org/10.1007/s11704-024-40555-y.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=xw5nxFWMlo.

Wang Xu, Kehai Chen, and Tiejun Zhao. Document-level relation extraction with reconstruction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14167–14175. AAAI Press, 2021b. doi: 10.1609/AAAI.V35I16.17667. URL https://doi.org/10.1609/aaai.v35i16.17667.

Wang Xu, Kehai Chen, Lili Mou, and Tiejun Zhao. Document-level relation extraction with sentences importance estimation and focusing. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2920–2929, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.212. URL https://aclanthology.org/2022.naacl-main.212/.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.365. URL https://aclanthology.org/2023.emnlp-main.365/.

Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1182/.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In Stefan Riezler and Yoav Goldberg, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1025. URL https://aclanthology.org/K16-1025/.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6575.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Wang, and Liangming Pan. How is LLM reasoning distracted by irrelevant context? an analysis using a controlled benchmark. *CoRR*, abs/2505.18761, 2025. doi: 10.48550/ARXIV.2505.18761. URL https://doi.org/10.48550/arXiv.2505.18761.

Xiaohan Yang, Eduardo Peynetti, Vasco Meerman, and Chris Tanner. What GPT knows about who is who. In Shabnam Tafreshi, João Sedoc, Anna Rogers, Aleksandr Drozd, Anna Rumshisky, and Arjun Akula, editors, *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 75–81, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.insights-1.10. URL https://aclanthology.org/2022.insights-1.10/.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1074. URL https://aclanthology.org/P19-1074/.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/f224f056694bcfe465c5d84579785761-Abstract-Conference.html.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. TextRunner: Open information extraction on the web. In Bob Carpenter, Amanda Stent, and Jason D. Williams, editors, *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA, April 2007. Association for Computational Linguistics. URL https://aclanthology.org/N07-4013/.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Xiang Chen, Hui Chen, Feiyu Xiong, Xi Chen, and Huajun Chen. Ontology-enhanced prompt-tuning for few-shot learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 778–787, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3511921. URL https://doi.org/10.1145/3485447.3511921.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. Give me the facts! a survey on factual knowledge probing in pre-trained language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore, December 2023. Association for Computational

Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1043. URL https://aclanthology.org/2023.findings-emnlp.1043/.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. Zero-shot temporal relation extraction with ChatGPT. In Dina Demner-fushman, Sophia Ananiadou, and Kevin Cohen, editors, *Proceedings of the 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 92–102, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.bionlp-1.7. URL https://aclanthology.org/2023.bionlp-1.7/.

Xinfeng Yuan, Siyu Yuan, Yuhan Cui, Tianhe Lin, Xintao Wang, Rui Xu, Jiangjie Chen, and Deqing Yang. Evaluating character understanding of large language models via character profiling from fictional works. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8015–8036, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.456. URL https://aclanthology.org/2024.emnlp-main.456/.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 71–78. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118703. URL https://aclanthology.org/W02-1010/.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In Junichi Tsujii and Jan Hajic, editors, *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://aclanthology.org/C14-1220/.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. Double graph based reasoning for document-level relation extraction. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.127. URL https://aclanthology.org/2020.emnlp-main.127/.

ChengXiang Zhai. Statistical language models for information retrieval: A critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, 2008. doi: 10.1561/1500000008. URL https://doi.org/10.1561/1500000008.

Hanlin Zhang, YiFan Zhang, Yaodong Yu, Dhruv Madeka, Dean Foster, Eric Xing, Himabindu Lakkaraju, and Sham Kakade. A study on the calibration of in-context learning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6118–6136, Mexico City, Mexico, June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.340. URL https://aclanthology.org/2024.naacl-long.340/.

Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. Document-level relation extraction as semantic segmentation. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3999–4006. ijcai.org, 2021. doi: 10.24963/IJCAI.2021/551. URL https://doi.org/10.24963/ijcai.2021/551.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models. *CoRR*, abs/2401.01286, 2024b. doi: 10.48550/ARXIV.2401.01286. URL https://doi.org/10.48550/arXiv.2401.01286.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. Bidirectional long short-term memory networks for relation classification. In Hai Zhao, editor, *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, Shanghai, China, October 2015. URL https://aclanthology.org/Y15-1009/.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1004. URL https://aclanthology.org/D17-1004/.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. Transfer learning from semantic role labeling to event argument extraction with template-based slot querying. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2627–2647, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.169. URL https://aclanthology.org/2022.emnlp-main.169/.

Runcong Zhao, Qinglin Zhu, Hainiu Xu, Jiazheng Li, Yuxiang Zhou, Yulan He, and Lin Gui. Large language models fall short: Understanding complex relationships in detective narratives. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7618–7638, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.454. URL https://aclanthology.org/2024.findings-acl.454/.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *CoRR*, abs/2303.18223, 2023. doi: 10.48550/ARXIV.2303.18223. URL https://doi.org/10.48550/arXiv.2303.18223.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Comput. Surv.*, 56(11):293:1–293:39, 2024b. doi: 10.1145/3674501. URL https://doi.org/10.1145/3674501.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 2021. URL http://proceedings.mlr.press/v139/zhao21c.html.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1032. URL https://aclanthology.org/D19-1032/.

Shun Zheng, Xu Han, Yankai Lin, Peilin Yu, Lu Chen, Ling Huang, Zhiyuan Liu, and Wei Xu. DIAG-NRE: A neural pattern diagnosis framework for distantly supervised neural relation extraction. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1419–1429, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1137. URL https://aclanthology.org/P19-1137/.

Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: Learning vs. learning to recall. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.398. URL https://aclanthology.org/2021.naacl-main.398/.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring various knowledge in relation extraction. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, editors, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219893. URL https://aclanthology.org/P05-1053/.

GuoDong Zhou, Min Zhang, Dong Hong Ji, and QiaoMing Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In Jason Eisner, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 728–736, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/D07-1076/.

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. A survey on neural open information extraction: Current status and future directions. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5694–5701. ijcai.org, 2022. doi: 10.24963/IJCAI.2022/793. URL https://doi.org/10.24963/ijcai.2022/793.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14612–14620. AAAI Press, 2021. doi: 10.1609/AAAI.V35I16.17717. URL https://doi.org/10.1609/aaai.v35i16.17717.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107, 2025. doi: 10.48550/ARXIV.2308.07107. URL https://doi.org/10.48550/arXiv.2308.07107.