Towards Practical

Automatic Document Understanding

MA YUBO

College of Computing and Data Science

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2025

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

11/03/2025
. . . .

Date

. . . .

MA YUBO

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

11/03/2025

Date

Sun Aixin

Authorship Attribution Statement

This thesis contains material from 6 papers accepted at the conferences in which I am listed as an author.

Chapter 2 is published as Shumin Deng, Yubo Ma, Ningyu Zhang, Yixin Cao, and Bryan Hooi. 2024. Information Extraction in Low-Resource Scenarios: Survey and Perspective. *IEEE International Conference on Knowledge Graph (ICKG)*, pages 33-49, Abu Dhabi, United Arab Emirates.

The author contributions are as follows:

- Shumin proposed the initial project, designed the experiments, wrote the first manuscript, and revised the manuscript.
- I designed the experiments, wrote the code and conducted all experiments.
- The other authors proofread the manuscript.

Chapter 3 is published as Yubo Ma, Zehao Wang, Yixin Cao, and Aixin Sun. 2023. Few-shot Event Detection: An Empirical Study and a Unified View. *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11211–11236, Toronto, Canada. Association for Computational Linguistics.

The author contributions are as follows:

- Prof. Sun and Prof. Cao suggested the research direction, and revised the manuscript.
- I came up with the idea, designed and performed most of the experiments, conducted data analysis, wrote the first manuscript, and revised the manuscript.
- Zehao performed some of the experiments, and revised the manuscript.

Chapter 4 is published as Yubo Ma, Zehao Wang*, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

The author contributions are as follows:

- Prof. Cao suggested the research direction, and revised the manuscript.

- I came up with the idea, designed and performed most of the experiments, conducted data analysis, wrote the first manuscript, and revised the manuscript.
- Zehao came up with the idea, performed some of the experiments, conducted data analysis, and revised the manuscript.
- The other authors proofread the manuscript.

Chapter 5 is published as Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!. *In Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601, Singapore. Association for Computational Linguistics.

The author contributions are as follows:

- Prof. Sun and Prof. Cao suggested the research direction, and revised the manuscript.
- I came up with the idea, designed and performed most of the experiments, conducted data analysis, wrote the first manuscript, and revised the manuscript.
- YongChing performed some of the experiments.

Chapter 6 is published as Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. MMLongBench-Doc: Benchmarking Long-context Document Understanding with Visualizations. *38th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, pages 95963-96010, Vancouver, Canada. Curran Associates, Inc.

The author contributions are as follows:

- Prof. Sun, Prof. Cao and Jiaqi suggested the research direction, and proofread the manuscript.
- I came up with the idea, supervised the annotation process, designed and performed most of the experiments, conducted data analysis, wrote the first manuscript, and revised the manuscript.
- Yuhang came up with the idea, co-supervised the annotation process, designed and performed some of the experiments, and revised the manuscript.
- Liangyu, Meiqi, Yizhu, Xinze, Xinyuan, Ziyu, Yan and Liangming annotated the benchmark, and proofread the manuscript.
- The other authors proofread the manuscript

Chapter 7 is published as Yubo Ma, Jinsong Li, Yuhang Zang, Xiaobao Wu, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Haodong Duan, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2025. Towards Storage-Efficient Visual Document Retrieval: An Empirical Study on Reducing Patch-Level Embeddings. *In Findings of the Association for Computational Linguistics: ACL 2025, pages 19568–19580, Vienna, Austria. Association for Computational Linguistics.*

The author contributions are as follows:

- Prof. Sun, Prof. Cao and Jiaqi suggested the research direction, and proofread the manuscript.
- I came up with the idea, designed and performed all of the experiments, conducted data analysis, wrote the first manuscript, and revised the manuscript.
- The other authors proofread the manuscript

11/03/2025

Date

MA YUBO

Acknowledgements

My Ph.D. journey was an unforgettable experience in my life. And I wish to extend my sincere and great appreciation to everyone who supported me throughout my Ph.D. journey.

First, it is my great honor to pursue my Ph.D. degree under the supervision of Prof. Sun Aixin and Prof. Cao Yixin. Their invaluable insights and guidances, strong encouragements and supports were indispensable for the completion of my four-year Ph.D. journey.

I also want to extend my sincere gratitude to my Thesis Advisory Committee (TAC) members, including Prof. Chen Lihui, Prof. Li Boyang and Prof. Joty Shafiq Rayhan. Thank you for your valuable comments and feedbacks on my research progress.

I am thankful to my collaborators, including Wang Zehao, Zang Yuhang, Li Xinze, Chen Meiqi, Pan Liangming, Chen Liangyu, Gou Zhibin, Deng Shumin, Jiao Yizhu, Li Mukai, Lu Xinyuan, and all others. Working with such brilliant individuals made my research experience truly enjoyable. I would also thank all my friends, colleagues, and labmates at NTU S-Lab and NTU CCDS.

Finally, I am deeply grateful to my parents for their boundless love. My heartfelt thanks go to my grilfriend, Ms. Li Linzhen. Their supports have been a source of strength throughout my research journey.

Abstract

Documents have been essential for information preserving and exchanging, with their significance continually increasing in the information age. Since the excessive documents make manual processing impractical, automatic document understanding has become a long-standing task with urgent and practical needs. Automatic document understanding enables machines to identify, extract, analyze, and reason over information from digital documents. Though being extensively studied, it faces several challenges: (1) **Few-shot**: The scarcity of annotated instances requires robust few-shot learning to capture rare but valuable information. (2) **Large-scale**: Efficient extraction and retrieval mechanisms are necessary to handle the explosive growth in document volume. (3) **Multi-modal**: Unified understanding demands integrated perception of diverse modalities like text, tables and charts. (4) **Long-context**: Accurate localizing and reasoning over information across lengthy documents remain an open problem. The above challenges hinder the practical deployment of document understanding systems.

Towards a practical automatic document understanding, this thesis addresses these challenges in two key sub-tasks: Information Extraction (IE) and Document Reading Comprehension. First, we start from **few-shot** event detection task and conduct an empirical study on previous competitive methods. Based on the results, we propose a unified view by breaking down the design elements of previous methods along several dimensions. Upon the unified view, we propose a simple yet effective unified baseline that combines all advantageous elements and performs best.

Second, we explore event argument extraction task and pursue the algorithm to be effective under **few-shot** settings and efficient under **large-scale** scenarios. Our proposed PAIE algorithm effectively captures the interactions of arguments scattered among documents, and efficiently extracts all of them in parallel. Extensive experiments show that PAIE presents promising improvements under few-shot settings and significant time reductions on large-scale documents.

Third, we investigate how to leverage recent Large Language Models (LLMs) into generalized **few-shot** and **large-scale** IE tasks. Our findings reveal that LLMs effectively complement Small Language Models (SLMs) and handle challenging samples that SLMs struggle with. Consequently, we propose an adaptive filter-then-rerank paradigm to combine the strengths of LLMs and SLMs, where SLMs make preliminary predictions and LLMs rerank a small portion of difficult samples identified by SLMs. Our paradigm achieves consistent improvements on various few-shot IE tasks, with acceptable latencies and budgets.

Fourth, we turn our attention to document reading comprehension and evaluate the reading comprehension capabilities of Large Language-Vision Models (LVLMs) on **multi-modal**, **long-context** documents. To this end, we construct a benchmark named MMLongBench-Doc incorporating lengthy PDF-formatted documents and questions based on these documents. The answers of these questions rely on evidences across different modalities and pages. Extensive experiments on 14 LVLMs reveal that the unified and end-to-end understanding of lengthy documents poses great challenges to current LVLMs.

Finally, we take a step back and pursue a visualized document retriever (VDR) which can retrieve key information from the **large-scale**, **multi-modal** documents. Identifying computational bottlenecks in existing VDR approaches, we propose a simple yet effective approach named Light-ColPali/ColQwen. It merges similar tokens to reduce both the offline memory cost and the online time latency with minimal performance loss. This approach makes VDR much more practical under real-world applications.

All in all, this thesis makes significant contributions to practical automatic document understanding in multiple dimensions. Through systematic investigation of both Information Extraction and Document Reading Comprehension, we conduct comprehensive empirical studies, establish new benchmarks, and develop innovative approaches to bridge the gap between theoretical research and real-world scenarios. We believe that these contributions lay a solid foundation for future research in document understanding.

Contents

List of Figures

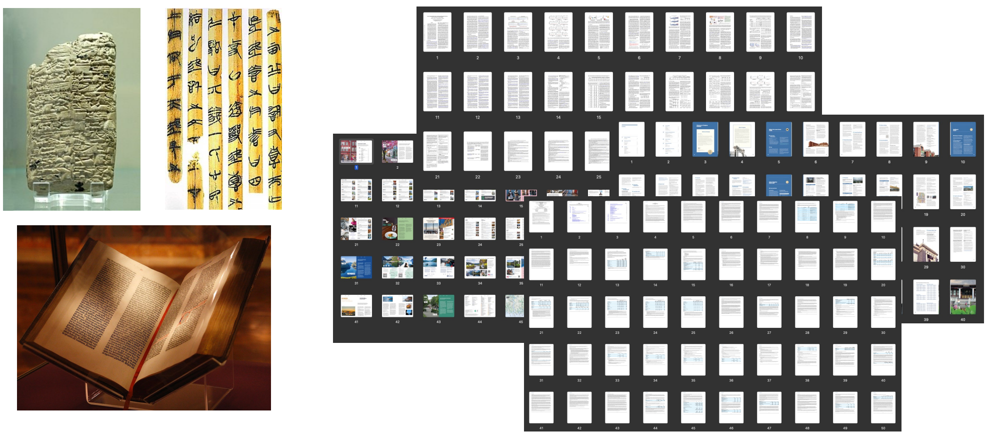
List of Tables

Chapter 1

Introduction

1.1 Motivations

Documents are semantically-coherent artifacts in which heterogeneous information units (sections, paragraphs, sentences, tables, figures, *etc.*) are aggregated into the whole. They are one of the fundamental forms for information preservation and sharing throughout human history. From early forms such as clay tablets, bamboos and papers shown in Figure 1.1(a), they have played a vital role in the progress of civilization. Over the past decades, the importance and prevalence of documents have further increased. At the same time, the paradigm of document creation, read, save, and dispatch has been significantly renewed.



(a) Ancient documents in different formats

(b) Digital documents in the information age

FIGURE 1.1: Document examples in (a) ancient times and (b) information age.

This shift in the information age can be summarized in two key aspects. Firstly, digitization makes documents beyond the physical formats (whether handwritten or printed) and enables virtual creation and storage. Documents now can be formatted as sequences of bits in electronic devices or cloud servers, such as PDFs, Word files, PowerPoint presentations, and scanned images of documents. It drastically reduces documentation costs and contributes to explosive growth of document amounts. Secondly, the popularity of Internet significantly decreases the barrier to document access and distribution. Unlike last generations searching for documents in libraries or archives, people can now readily access extremely abundant documents online. Consequently, these two developments, *i.e.,* the rapid growth of and easy access to documents, have resulted in a unprecedentedly rich collection of accessible documents.

The wealth of accessible documents greatly facilitate the information recording and sharing in human society. When facing millions of or even billions of documents, however, conventional manual reading becomes time-consuming, costly, and somewhat impractical. Additionally, the rapid progress of General Artificial Intelligence (GAI) draws a promising illustration that AI-driven systems will likely reach or even surpass the limit of humans in the very near future. Therefore, the automatic processing of documents can benefit wide applications across various domains, particularly in areas where timely decision-making and scalable information-seeking are critical such as legal analysis [3, 4], financial analysis [5, 6], medical record interpretation [7], academic research [8, 9], *etc.*

1.2 Research Scope

Automatic document understanding enables machines to identify, extract, analyze, and reason over the semantics of digital documents and/or their parsed units. In this thesis, we focus on automatic document understanding and examine two representative sub-tasks, *i.e.,* Information Extraction and Document Reading Comprehension. Each of them addresses practical needs in document understanding with different formats and levels. We detail these two sub-tasks as follows.

Document understanding as information extraction: [1] When the needs are clear, an extractive approach called information extraction (IE [10]) is often used to obtain specific contents from documents. For example, financial analysts extract key metrics like net profit and debt-to-asset ratios from annual reports; doctors quickly locate specific test results in electronic medical records; and editors locate information related to a specific event from numerous news articles. As shown in Figure 1.2 (left), IE identifies and extracts structured knowledge from document units . Here are two key concepts in IE task. (1) The schema specifies the types and classes of information to be extracted, such as entities [11], relations [12], events [13], *etc.* (2) The structured knowledge is derived from the document's context and formatted as the schema's framework. By transforming unstructured text into structured data, IE serves as an intermediate task in document understanding by summarizing the key information from large document corpora in concise and organized manners. It makes the information in the documents easier to be analyzed and utilized, and benefits various applications such as knowledge-base construction [14–17], question answering [18, 19] and story generation [20, 21].

Document understanding as reading comprehension: When the needs are more personalized and/or complicated, the reading comprehension is more attractive. For instance, researchers summarize the methodology, theoretical framework, and innovations of academic papers; lawyers assess the legal validity and potential risks of contracts. Featuring flexible inputs and outputs, Reading Comprehension [22] is formatted as a question answering (QA) problem and somewhat represents a recent paradigm shift in document understanding. As shown in Figure 1.2 (right), this task receives the question and provides the answer based on either the document units or the whole documents. Compared to IE tasks, reading comprehension understands documents in a more flexible and dynamic manner. Here, the questions and answers are written in natural languages and not constrained by rigid formats or predefined templates. It enables them to meet diverse and customized

[1] We argue that Information Extraction (IE) is one of the critical tasks for document understanding (and most general NLP tasks are not). IE is designed to extract and structuralize the key information from the documents. However, due to the limitations of computational cost and/or model architectures, most IE systems process documents at sub-document level. They separate documents into multiple units and process each chunk independently. Although a unit may contain only one or two sentences, the structured information extracted at sub-document levels is ultimately co-referenced and/or aggregated at the whole-document level. These aggregated information further serves for downstream document-centric use, such as search, analytics, QA, etc. In contrast, most other general NLP tasks neither integrate information across multiple units of the same document, nor serve the broader goal of document understanding in this sense.

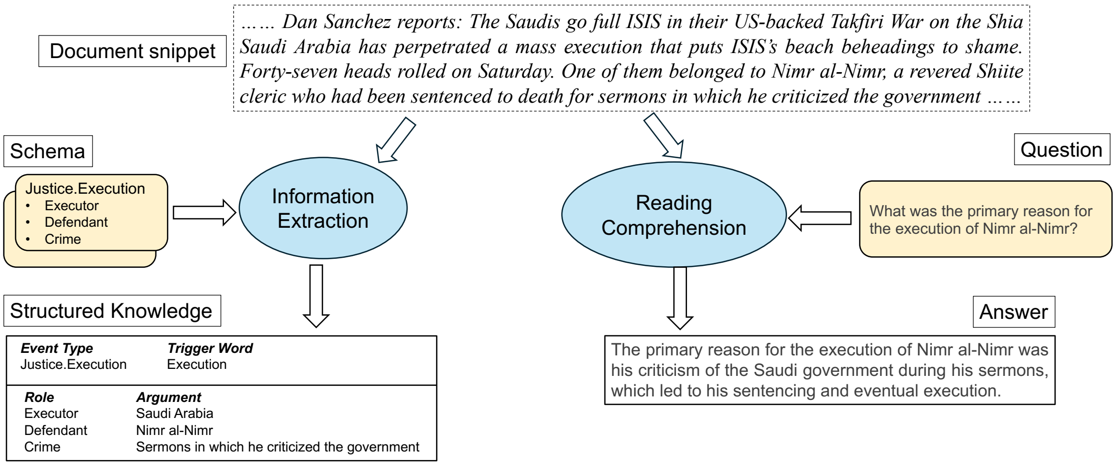


FIGURE 1.2: Illustration of two focused sub-tasks of automatic document understanding in this thesis. **Left**: Information Extraction. **Right**: Document Reading Comprehension.

demands ranging from low-level questions [23, 24] (extracting information; *e.g.,* What is the publication date of this document?) to high-level questions [25, 26] (summarizing or reasoning upon deep analysis; *e.g.,* What are the main arguments presented in this paper, and how do they compare to prior work?). With the surge development of Large Language Models (LLMs), the power of reading comprehension has been more and more leveraged under real-world scenarios.

1.3 Challenges

Even though automatic document understanding has been intensively studied, significant challenges remain in the real-world applications. In this section, we summarize the challenges as follows. These discussed challenges motivate us to develop targeted works in the later sections of this thesis.

- **Few-shot**. The information and knowledge in documents exhibit a long-tailed distribution [27]. While certain types of information appear frequently and can be easily extracted, many other types rarely occur and are challenging to extract. In practice, most classes of the IE schema fall into the tail category and have few instances in document corpora. This scarcity is further exacerbated by the fact that annotating structured instances (such as entities, relationships, or events) is notoriously difficult, time-consuming, and often requires domain expertise. As a result, the training samples for most classes in IE tasks are scarce. In such low-resource scenarios, extractors must accurately identify schema types that occur only a few times or even zero times.

- **Multi-modality**. Raw documents (such as PDFs, Word files, PowerPoint presentations, and scanned images of documents) are multi-modal in native and feature complex layouts and heterogeneous information including text, tables, charts, images, and more. Though document understanding systems are expected to handle raw documents towards a unified comprehension, they currently still struggle with integrating diverse modalities effectively. Text-based models may excel at processing paragraph but fail to interpret tables or images. Conversely, vision-based models might capture visual elements but miss the fine-grained textual details. How to handle information across different modalities, *i.e.,* perceiving both the textual paragraphs and the visualized components in the document, poses great challenges for the current document understanding systems.

- **Long-context**. Considerable amounts of documents in the real world are long-context documents with tens or even hundreds of pages. The understanding of these lengthy documents brings additional challenges from at least two aspects: (1) Localization: accurately identifying information from massive information; (2) Cross-page comprehension: collecting and reasoning over multi-source information across different pages. Even though the document systems empowered by Large Language Models (LLMs; [28]) and Large Vision-Language Models (LVLMs [29]) can receive inputs with million tokens, they still struggle on these above two challenges [30, 31].

- **Large-scale**. As mentioned earlier, the explosive growth of document amounts is a key driver for automated understanding. When dealing with large-scale documents, two major challenges arise. First, we need efficient and scalable IE algorithms that can quickly and cost-effectively extract structured knowledge. Second, the specific page or document containing the key information is usually unknown. Since it is impractical to process all documents in a corpus to answer a single question, a efficient retrieval system which selects relevant documents or even pages from massive corpora is in high demands.

1.4 Contributions

In this section, we summarize our contributions that aim to address the challenges mentioned above. We categorize the discussion into the part of *Information Extraction* and *Reading Comprehension*, respectively.

Information Extraction: The first part of this thesis introduces three IE algorithms which are effective under **few-shot** settings and efficient under **large-scale** scenarios. The first two works focus on two event-related IE tasks, *i.e.,* event detection and event argument extraction. And the last work is generalized across different IE tasks.

- We start from event detection task and conduct an empirical study of twelve SOTA methods under practical **few-shot** settings. Based on the result, we propose a unified view by breaking down the design elements of previous methods along several dimensions. Upon the unified view, we propose a simple yet effective unified baseline that combines all advantageous elements of existing methods and performs best. This work is published in the Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023) [32].

- We further explore the applications of prompting strategy for IE task and propose a novel method PAIE that prompting argument interactions for event argument extraction task. Our prompting strategy effectively captures the interactions of arguments scattered among documents, and efficiently extracts all arguments in parallel. Extensive experiments show that PAIE presents promising performance improvement under **few-shot** settings and significant time reduction on **large-scale** documents. This work is published in the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022) [33].

- The emergence of Large Language Models (LLMs) brings new opportunities and possibilities towards **few-shot** IE tasks [34, 35]. Thus we explore the optimal strategy to leverage LLMs by the means of in-context learning. Our findings reveal that LLMs are not independent effective few-shot information extractors but can effectively complement Small Language Models (SLMs) and tackle challenging samples that SLMs struggle with. Consequently, we propose an adaptive filter-then-rerank paradigm to combine the strengths of LLMs and SLMs. By prompting LLMs to rerank a small portion of difficult samples identified by SLMs, our preliminary system consistently achieves promising improvements on various **few-shot** IE tasks, with an acceptable time and cost investment on **large-scale** documents. This work is published in the Findings of the Association for Computational Linguistics: EMNLP 2023 (EMNLP 2023; Findings) [36].

Document Reading Comprehension: The second part of this thesis focuses on two related works about reading comprehension. Here the document refers to the native **multi-modal** documents such as PDFs, PowerPoints, *etc.* Specifically, the first work explores end-to-end reading comprehension for **long-context** documents, while the second work investigates the efficiency of a visualized document retriever towards **large-scale** applications.

- We aim to evaluate the reading comprehension capabilities of recent Large Language-Vision Models (LVLMs; [29]) on **multi-modal**, **long-context** documents. To this end, we construct MMLongBench-Doc incorporating lengthy PDF-formatted documents and questions based on these documents. The answers of these questions rely on evidences across different modalities and pages. Extensive experiments on 14 LVLMs reveal that the understanding of lengthy documents poses great challenges to current LVLMs. To our best knowledge, MMLongBench-Doc is the first comprehensive benchmark designed to evaluate the long-context document understanding capabilities of LVLMs. This work is published in the 38th Conference on Neural Information Processing Systems as a spotlight paper (NeurIPS 2024; Spotlight) [37].

- We further explore the Visualized Document Retrieval (VDR; [38, 39]) task and aim to retrieve short document snippets from **multi-modal**, **large-scale** document corpora effectively and efficiently. Specifically, we conduct an empirical study to pursue the optimal efficient retriever which reduces maximum offline memory footprints at minimum performance degradation. Our findings reveal that *token merging* is naturally more appropriate than *token pruning* under VDR setting. And we search for the optimal combinations of merging strategies across three dimensions. Based on the searching results, we combine advantageous choices and develop Light-ColPali/ColQwen2

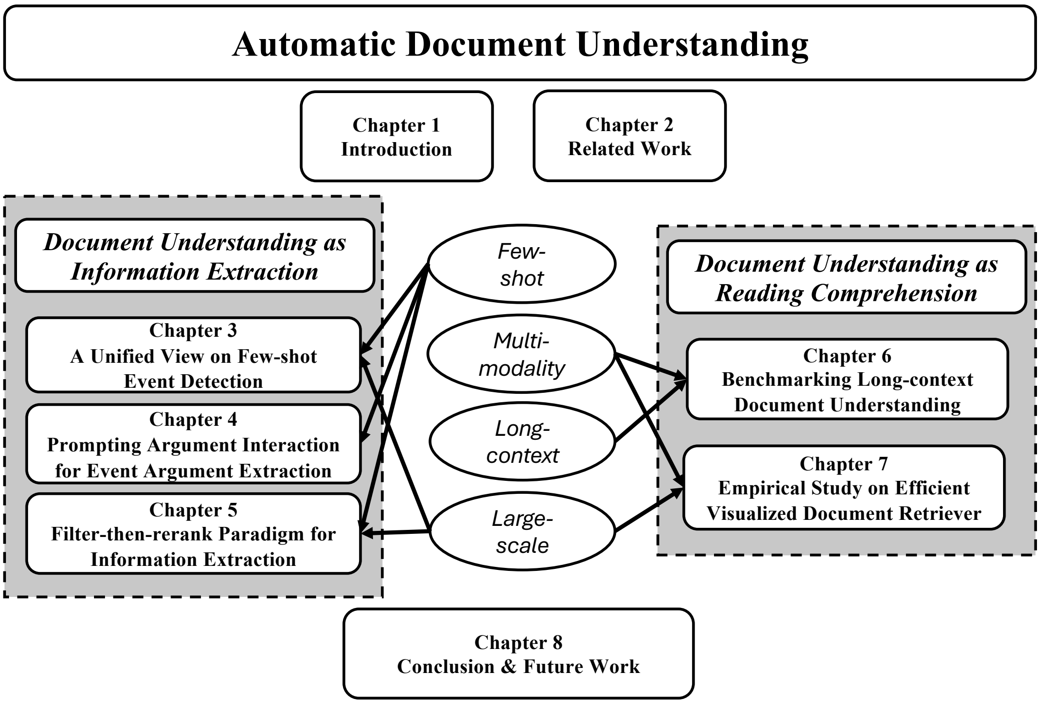


FIGURE 1.3: The outline of this thesis. Rectangles represent the chapters and circles represent the challenges to address.

which reduces the embedding footprints by orders of magnitude and maintains comparable performance. This work is published in the Findings of the Association for Computational Linguistics: ACL 2025 (ACL 2025; Findings) [40].

1.5 Thesis Outlines

As illustrated in Figure 1.3, this thesis is outlined in the following manner:

Chapter 1 introduces the motivations of automatic document understanding. It figures out four practical challenges in document understanding, and summarizes our contributions to its two key sub-tasks: information extraction and reading comprehension. **Chapter 2** then provides a literature review on these tasks.

Chapter 3-**Chapter 5** focus on three few-shot and efficient information extraction algorithms. Specifically:

- **Chapter 3** conducts an empirical study on few-shot event detection. It proposes a unified view of previous works and develops a baseline that combines the strengths of existing methods. This baseline outperforms all previous approaches.

- **Chapter 4** introduces PAIE, a prompt tuning strategy that extracts all arguments at once and captures their interactions. To our knowledge, it is the first prompt-based method for event argument extraction achieving strong performance.
- **Chapter 5** explores the optimal use of LLMs for few-shot IE tasks. Take a step further, we propose a filter-then-rerank paradigm that combines traditional IE algorithms with LLM-based approaches.

Chapter 6-Chapter 7 address the reading comprehension task for multi-modal documents. Specifically:

- **Chapter 6** focuses on long-context reading comprehension task and, to our best knowledge, presents MMLongBench-Doc benchmark for this domain. Experiments reveal that the understanding of lengthy documents poses great challenges to current LVLMs.
- **Chapter 7** investigates the efficiency of visualized document retrievers. We propose a token merging approach that achieves comparable accuracy while reducing memory costs by over 90%.

Chapter 8 concludes the thesis and discusses the potential directions for future research towards more practical document understanding.

Chapter 2

Literature Review

In this chapter, we review existing works on automatic document understanding. Following the categorization in Chapter 1, we discuss document understanding from two aspects, *i.e.,* Information Extraction and Document Reading Comprehension, in Section 2.1 and Section 2.2 respectively.

2.1 Information Extraction

In this section [1], we begin with the definition of four Information Extraction (IE) tasks. Next, we clarify the few-shot IE setting discussed in this thesis. Then we review few-shot IE approaches on Small Language Models (SLMs; like BERT [41], BART [42], T5 [43], *etc.*) and Large Language Models (LLMs; like GPT-4 [44], CODEX [45], LLaMA [46], *etc.*) respectively.

Task Definition. Generally, Information Extraction (IE; [47, 48]) can be regarded as structured prediction tasks [49], where a classifier is trained to approximate a target function $F(x) \to y$. Here $x \in X$ denotes the input data and $y \in Y$ denotes the output class sequence. This thesis involves four sub-tasks of IE:

- **Named Entity Recognition** (NER [11]) locates and classifies named entities from the documents. It is usually formulated as a sequence labeling task.

[1] This section is published as Shumin Deng, Yubo Ma, Ningyu Zhang, Yixin Cao, and Bryan Hooi. "Information Extraction in Low-Resource Scenarios: Survey and Perspective". IEEE International Conference on Knowledge Graph (ICKG), pages 33-49, Abu Dhabi, United Arab Emirates, 2024. [10]

Given a document corpora D annotated with schema E (the set of entity types) and a sentence $X = [x_1, ..., x_N] \in D$, where x_i is the i-th word and N the length of this sentence, NER aims to assign a label $y_i \in (E \bigcup \{N.A.\})$ for each $x_i \in X$. Here $N.A.$ refers to either none entities or entities beyond pre-defined types E.

- **Relation Extraction** (RE [50]) classifies the relations between entity pairs and is usually formatted as a classification task. Given an entity pair $(e_1, e_2) \in E \times E$ and a relation set R, RE aims to assign a relation label $y_i \in (R \bigcup \{N.A.\})$ to the entity pair.

- **Event Detection** (ED [13]) locates and identifies the event triggers and their types occurred in the documents. It shares similar task format as NER and can be formulated as a sequence labeling task. The only difference is that ED extracts event triggers $x_i \in X$ about event type $e \in E$. Here schema E denotes the set of event type (rather than entity type).

- **Event Argument Extraction** (EAE [13]) is a successive task of ED which locates and identifies the arguments of the specific event type and its trigger word. EAE is formatted as a span extraction problem on document D. Given an instance $(D, t, e, R(e))$ where $t \subseteq D$ denotes the trigger word, e denotes the event type and $R(e)$ denotes the set of event-specific role types, we aim to extract a set of span A. Each $a(r) \in A$ is a segmentation of D and represents an argument about $r \in R(e)$.

For instance, given a sentence

Jack is married to the Iraqi microbiologist known as Dr. Germ.

NER should identify the types of entities in given texts, *e.g., Jack* and *Dr. Germ* as entity `PERSON`; RE should identify the relationship of the given entity pair ⟨*Jack*, *Dr. Germ*⟩ as relation `husband_of`; ED should identify the event type as `Marry`, where the word *married* triggers the event; EAR should further identify *Jack* and *Dr. Germ* as two arguments of role `husband` and role `wife`, respectively.

Few-shot IE setting. This thesis does **NOT** adopt the traditional few-shot task setting, *i.e.*, episode learning shown in Figure 2.1 (left). Literally, episode learning incorporates multiple episodes. Each episode is a few-shot problem with its own train (support) and test (query) sets and event-type classes. Since the sets in each episode are sampled uniformly having K different classes and each class having N instances, episode learning is also known as N-way-K-shot classification. We argue that this setting assumes a realistic scenario. First, during the meta-training stage, a large number of episodes is needed, for example, 20,000 in Cong et al. [51]. Second, tasks with episode learning are evaluated by the performance on samples of the test (query) set in the meta-testing phase. The test sets are sampled uniformly, leading to a significant discrepancy with the true data distribution in many NLP tasks. The absence of sentences without any events further leads to distribution distortion. Further, each episode contains samples with only K different classes, where K is usually much smaller than the event types in the target schema. All these factors may lead to an overestimation on the ability of few-shot learning systems. For above reasons, we do not consider this setting in our experiments.

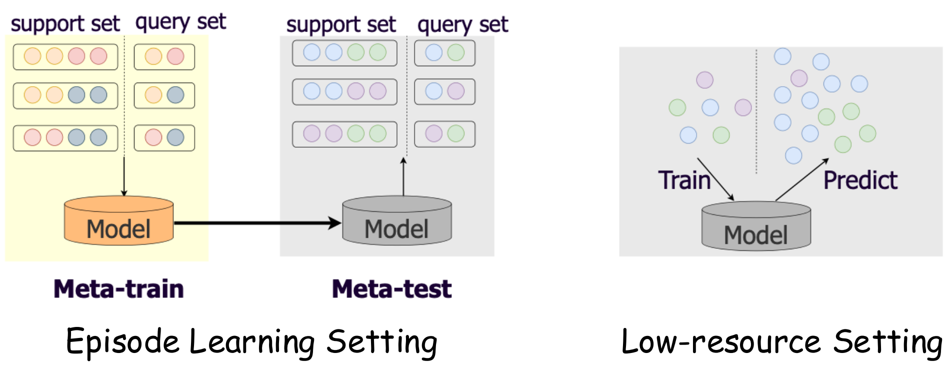


FIGURE 2.1: Illustration of two few-shot settings. Different colors represent different event types. Different colors represent different classes in the schema (*e.g.*, event types for event detection task). We adopt the right one in this thesis.

Instead, we consider a more simple and practical few-shot setting shown in Figure 2.1 (right). It is also known as low-resource setting. Under this setting, we assume access to a dataset $\mathcal{D} = (\mathcal{D}_{train}, \mathcal{D}_{dev}, \mathcal{D}_{test})$ annotated with a label set E, where $|\mathcal{D}_{dev}| \leq |\mathcal{D}_{train}| \ll |\mathcal{D}_{test}|$. It assesses the generalization ability of models by (1) utilizing only few samples during training, and (2) evaluating on the real and rich test dataset.

2.1.1 Few-shot IE Approaches on SLMs

We introduce previous few-shot approaches on SLMs categorized by their corresponding IE tasks as follows.

- **Named Entity Recognition** One classical paradigm in NER is utilizing ProtoNet [52] and its variants to learn *one* representative prototypes for each class type with only few examples. Fritzler et al. [53] firstly combine ProtoNet and CRF module to solve NER tasks. Hou et al. [54] propose L-TapNet-CDT, which enhances TapNet [55], a variant of ProtoNet, with entity names and achieves great performance. Both methods construct prototypes by computing the average embeddings of several sampled examples (support set). Yang and Katiyar [56] propose a simpler algorithm, leveraging supervised classifier learned in existing schema as feature extractor and adopting nearest neighbors classification during inference, and show competitive performance in class transfer setting for few-shot NER task. Das et al. [57] introduce contrastive learning into few-shot NER task. Ma et al. [58] recently developed a simple but effective method on few-shot NER by constructing prototypes only with their labels.

- **Relation Extraction** We roughly categorize existing few-shot RE methods into two classes: prompt-based methods and prototype-based methods. (1) *Prompt-based methods* leverage the rich language knowledge in PLMs by converting downstream tasks to the task with which PLMs are more familiar. Such format conversion narrows the gap between pre-training and downstream tasks and benefits knowledge induction in PLMs with limited annotations. Specifically, few-shot RE can be converted to machine reading comprehension (MRC [59, 60]), natural language inference (NLI [61]) and summarization [62]. (2) *Prototype-based methods* predict an event type for each word/span mention by measuring its representation proximity to *prototypes* [63–67]. Here we define prototypes in a *generalized* format — it is an embedding that represents some event type.

- **Event Detection** Similarly as few-shot RE, we group existing few-shot ED methods into two classes as follows. (1) *Prompt-based methods.* Few-shot ED can be converted to MRC [68–70], NLI [71], conditional generation [72–75], and the cloze task [76]. (2) *Prototype-based methods.* For example, Prototypical Network (ProtoNet, [52]) and its variants [51, 77–81] construct prototypes via a subset of sample mentions. In addition to event mentions, a line of work leverage related knowledge to learn or enhance prototypes' representation, including AMR graphs [82], event-event relations [80], definitions [83] and FrameNet [84]. Zhang et al. [85] recently introduce contrastive learning [86] in few-shot ED task. Such method also determines the event by measuring the distances with other samples and aggregates these distances to evaluate an overall distance to each event type.

- **Event Argument Extraction** A recent trend formulates few-shot EAE as an extractive QA problem [68, 69]. This paradigm naturally induces the language knowledge from pre-trained language models by converting EAE tasks to fully-explored reading comprehension tasks via a question template. Wei et al. [87] considers the implicit interaction among roles by adding constraints with each other in template, while Liu et al. [88] leverages data augmentation to improve the performance. However, they can only predict roles one by one, which is inefficient and usually leads to sub-optimal performance. There are also some recent works converting extraction tasks to generation tasks. Paolini et al. [72] propose TANL to handle a variety of structured prediction tasks, including EAE, by a unified text-to-text approach and extract all arguments in a single pass. Lu et al. [73] follow TANL and also take EAE as a sequential generation problem. Li et al. [89] target generation model by designing specific templates for each event type.

2.1.2 Few-shot IE Approaches on LLMs

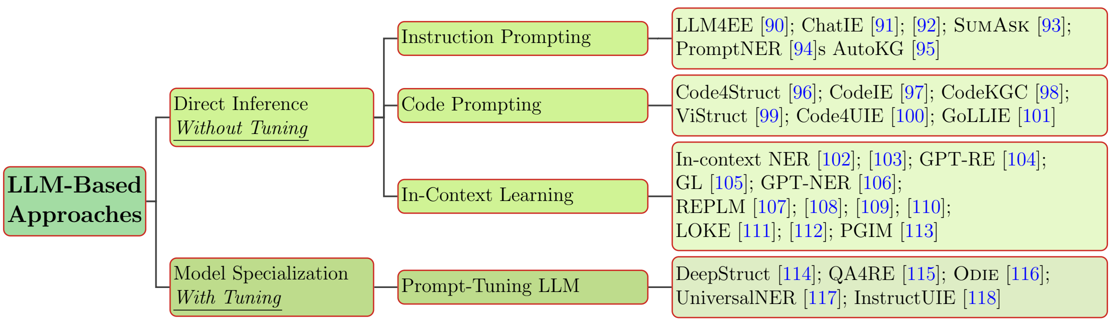


Figure 2.2: Taxonomy of few-shot IE approaches on LLMs.

Compared to SLMs, LLMs have stronger capabilities and allow for more advanced prompt learning strategies. Additionally, they also use a more standardized input-output format and allow for the review of few-shot IE methods from a unified perspective. We divide LLM-based approaches into two classes shown in Figure 2.2.

Direct Inference without Tuning. These methods prompt LLMs without fine-tuning and thus are practical for extracting information from scarce data. By leveraging the inherent capabilities of LLMs to understand the document, they enable LLMs to conduct few-shot IE tasks smoothly.

- **Instruction Prompting** provides task-specific instructions to the LLM. Since this paradigm requires no demonstrations, it is well-suited for zero-shot [91, 95, 119] and cross-domain [94] IE tasks. For example, Xie et al. [120], Li et al. [93] and Gao et al. [90] highlight LLMs' strengths in zero-shot named entity recognition (NER), relation extraction (RE), and event detection tasks, respectively. Additionally, Lin et al. [92] apply global constraints with prompting for zero-shot event extraction (EE) and show its adaptability across different datasets. However, there still exists remaining challenges that the prompts often struggle to clarify complex IE tasks or fine-grained schema types.

- **In-Context Learning (ICL)** leverages the ability of LLMs to learn from demonstrations. By understanding relevant examples, LLMs "understands" the specific IE task. Recent studies have applied ICL to various few-shot IE tasks, including NER [102, 106, 108, 112, 113], RE [103, 104, 107, 121], joint IE [105, 109], and OpenIE [110, 111]. However, the alignment between inputs and labels is not always effective. And we hope this can be further explored in the future work.

- **Code Prompting** provides LLMs with code-like instructions [101] to guide their response generations. Since the schema of IE tasks can be naturally formatted as code, this approach is particularly effective for few-shot IE tasks. For instance, Wang et al. [96], Li et al. [97], Bi et al. [98] design unified code-style prompts across various IE tasks. Guo et al. [100] introduce a retrieval-augmented code generation framework which integrates ICL (*i.e.*, retrieving similar examples) to code prompting. Additionally, Chen et al.

[99] extend code prompting to multi-modal scenarios and combine code and visual representations for visual IE tasks.

Model Specialization with Tuning. These methods improve few-shot IE by adjusting LLMs' weights on domain-specific data. We further categorize them into two approaches, *i.e.,* prompt-tuning and fine-tuning, and detail them as below.

- **Prompt-Tuning LLM** keeps most of the weights fixed and tunes a small set of parameters. Drawing inspiration from instruction tuning, several methods transform annotated samples into instruction-answer pairs and then finetune LLMs on them. For example, Wang et al. [118] propose a unified IE framework named Instruct-UIE which unifies 32 diverse IE datasets in a text-to-text format and performs instruction tuning on FlanT5-11B [122]. Jiao et al. [2] create InstructIE dataset and fine-tune LLaMA-7B [46] to enhance its instruction-following capabilities. Wang et al. [114] pre-train GLM-10B [123] on task-agnostic corpora to generate structured outputs from documents. Zhang et al. [115] figure out that instruction-tuning alone may not fully unlock RE capabilities in LLMs. To address this limitation, they propose QA4RE framework and align RE with question answering task (with which LLMs are more familiar). Above efforts highlight the diversity and effectiveness of prompt-tuning in adapting LLMs across generalized IE tasks.
- **Fine-Tuning LLM** adjusts the full weights on task-specific datasets. This approach is more data-intensive than prompt-tuning but expectantly leads to performance gains. Actually, this kind of methods are currently underdeveloped for low-resource IE tasks due to the limit of computing resources.

2.2 Document Reading Comprehension

In this section, we first define Document Reading Comprehension (DRC). We then briefly discuss the differences between textual-based and visual-based document reading comprehension. To tackle the *multi-modality* challenge outlined in Section 1.3, this thesis focuses on visual-based document reading comprehension. Specifically, we review two paradigms for (visual-based) reading comprehension:

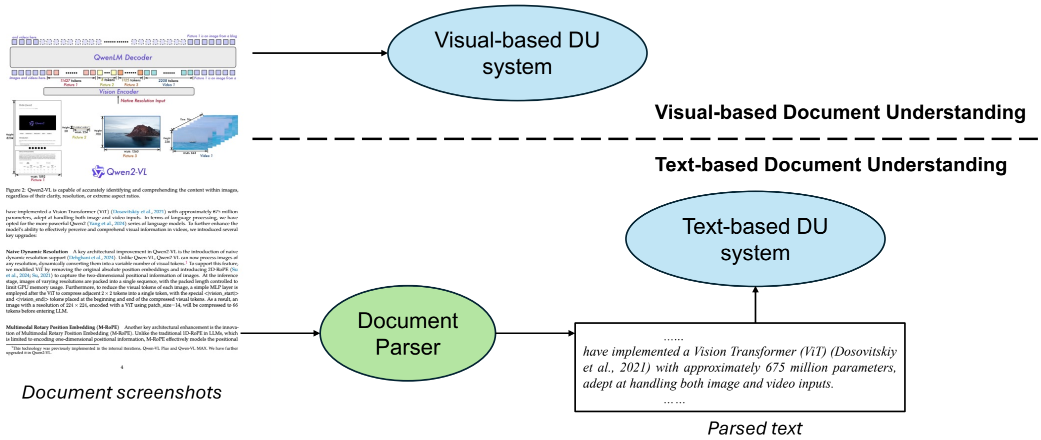


FIGURE 2.3: The illustration of two reading comprehension systems categorized by modality. **Top**: Visual-based. **Bottom**: Text-based.

end-to-end generation in Section 2.2.2 and retrieval-augmented generation in Section 2.2.3 respectively.

Generally, Document Reading Comprehension [23–25, 124] features more flexible manner and is formatted as a question answering (QA) task. Given a document context D [2] and a related question X, document reading comprehension aims to find the evidences from D to answer the question Y. This process emphasizes identifying and reasoning over key information within the document to address the question effectively.

2.2.1 Textual v.s. Visual Reading Comprehension

As shown in Figure 2.3, we classify reading comprehension into two categories based on the modality of D. If D is in text form, the system is categorized as text-based. If D is in captured as a screenshot and formatted as an image, the system falls under visual-based setting. We further clarify their differences as follows.

Text-based Reading Comprehension firstly parses text from the document and then feeds the text to the system. For example, the text can be parsed and curated from Wikipedia [23, 124–126], Reddit [127], Stack Exchange [128], Arxiv [129, 130], financial reports [5, 131, 132], *etc.* Text-based setting has historically dominated the field for two main reasons: **(1)** The NLP techniques, including the recent rise

[2] Here the context can be one or multiple text chunks, one or multiple paragraphs, even one or multiple complete documents.

of LLMs, have enabled robust understanding of parsed text. **(2)** The sheer scale of documents in the information age makes text-based storage significantly more cost-effective, as textual data naturally contains more information than pixel values. However, text-based setting also has intrinsic limitations compared to the visual-based approaches because it requires document parsers to extract the text from native documents. However, such document parsing is time-consuming and leads to significant latency under real-time scenarios. And it often results in information loss due to the incorrect recognitions on texts or tables. Moreover, it cannot utilize information from visual elements like figures, charts and diagrams.

Visual-based Reading Comprehension views documents as images (screenshots per page) and directly feeds the screenshots to the system. This approach preserves both the layout structures and visual elements of documents, enabling a what-you-see-is-what-you-get (WYSIWYG) understanding. Although it has been extensively studied [133–135], visual-based reading comprehension systems have historically underperformed than text-based ones. We mainly attribute this gap to the limited capabilities of visual models in processing large amounts of text within images, and their inability to perform deeper understanding and reasoning. However, recent LVLMs have demonstrated remarkable OCR [136] and reasoning [137, 138] capabilities. These developments highlight the growing potential of visual-based reading comprehension in evaluation, algorithms, and applications. As a result, we focus on visual-based setting in the second part of our thesis (Chapter 6-7) for detailed exploration.

Similar to text-based setting, visual-based reading comprehension can be further divided into two paradigms. The first paradigm, referred to as *end-to-end generation*, processes the entire document and generates the answer in a single step. The second paradigm, known as *retrieval-augmented generation*, is used when dealing with large document corpora (*e.g.,* Wikipedia) or when a single document is excessively long. We discuss related works on these two paradigms in Section 2.2.2 and Section 2.2.3, respectively.

2.2.2 End-to-end Visual Reading Comprehension

Models. End-to-end visual reading comprehension are primarily addressed by two main branches of models. The first branch employs two-stream, OCR-dependent architectures that separately encode textual information (extracted via OCR) and visual information (images and/or layout structures). Examples include the LayoutLM v1-v3 series [139–141], DocLLM [142] and Arctic-TILT [143]. The second branch eliminates the dependency on OCR and enables models to process image-formatted documents in an end-to-end manner. Notable examples include Donut [144], UODP [145], and Pix2Struct [146]. By incorporating large amounts of OCR data during training, recent LVLMs [147–151] have demonstrated strong performance on reading comprehension tasks where the documents are single-page or short-context. Accordingly, this branch has become the dominant approach for visual-based reading comprehension.

Benchmarks. A great amount of datasets have emerged to evaluate the reading comprehension capabilities of LVLMs. Previous datasets primarily focus on specific components (*e.g.*, tables, charts) [130, 152–154] or single-page documents [24, 155]. Recent efforts attempt to assess multi-page document reading comprehension, but still exhibit shortcomings in terms of document length (number of pages), information density (number of tokens), and construction methodologies. For instance, MP-DocVQA [156], an extension of DocVQA [24], lacks cross-page and unanswerable questions. DUDE [157], annotated from scratch, includes a small proportion of cross-page questions (2.1%) and unanswerable questions (12.7%). However, its relatively short context length and reliance on crowd-sourced annotations result in less challenging and less rigorous questions. SlideVQA [158] features 20-page documents and cross-page questions (12.9%). Nevertheless, their documents are in slide-deck format and of relatively low information density. Moreover, these cross-page questions are HotpotQA-style [25] created by instantiating entity graphs and co-referencing in-graph entities across multiple pages. Such construction approaches likely lead to false cross-page questions that actually do not require answers evidence across different pages. The recent FinanceBench [159] features both extremely long-context documents and practical, scalable cross-page questions. However, its documents are exclusively financial reports. Additionally, the reference answers are in open-ended formats, making the expert-level manual evaluation indispensable. The above reasons limit the broader applicability of FinanceBench. In summary, there remains a lack of a comprehensive and robust benchmark for evaluating long-context document understanding tasks.

Long-context LVLMs. Lengthy documents necessitate the use of LVLMs with extended context sizes. Several benchmarks [160–163] and solutions [164–167] have been proposed to evaluate and develop long-context LLMs. However, there exists limited related work for long-context LVLMs, leaving this area largely unexplored. Until very recently, contemporary studies [168–170] assess and/or improve LVLMs' multi-image understanding capabilities. Evaluations indicate that current LVLMs are not yet fully equipped to handle long-context reading comprehension (and many other practical tasks that require extensive contextual comprehension).

2.2.3 Retrieval-augmented Generation for Visual Reading Comprehension

Retrieval-augmented generation (RAG [171]) serves at least two critical roles in visual-based reading comprehension. (1) Current LVLMs still struggle with lengthy documents. By retrieving and processing only the most relevant sections of a document, RAG mitigates the limitations of handling long-context inputs. Therefore, it provides a temporary solution until more advanced long-context LVLMs are developed. (2) When dealing with large document corpora (*e.g.,* the entire Wikipedia), it is impractical to feed all documents into LVLMs. Retrieving the most relevant documents or sections is necessary in such scenarios. This approach ensures efficiency and scalability while maintaining the quality of comprehension.

An RAG-based document system typically consists of visual document retrievers (VDR) and LVLMs. By retrieving the most relevant document sections, VDR enables LVLM to handle shorter, more relevant documents. And the shortening of context enables LVLMs to generate more accurate and appropriate answers. While the role of LVLMs has been discussed in the previous section, this section focuses the review on VDR.

Visual Document Retriever. Given a user query q and a corpus of M-sized documents D (where $|D| = M$), VDR system aims to retrieve the top-k relevant subset of documents $D_q \subseteq D$ with $k \ll M$. VDR shares a similar architecture with text-based dense retrievers [172, 173], but leverages LVLMs [149, 174] for OCR-free document understanding. It primarily divides into two approaches:

(1) *Page-level embedding retrievers* encode entire pages and queries into single embeddings. Examples include DSE [39], VisRAG-Ret [175] and GME [176]. Given

the query q with tokens N_q and the document-formatted image p with patches N_p, it encodes them as query embeddings $e_q \in R^d$ and page embedding $e_p \in R^d$ in a unified embedding space using the LVLM backbone. The relevance score between q and p, denoted as $s(q, p)$, is calculated by their cosine similarity.

$$\begin{aligned} e_p &= \textbf{Retriever}(p) \in R^d \\ e_q &= \textbf{Retriever}(q) \in R^d \\ s(q, p) &= e_p^T e_q \end{aligned}$$

(2) *Patch-level embedding retrievers* like ColPali/ColQwen2 [38] generate multiple patch-level embeddings per page and token-level embeddings per query. Given query q with tokens N_q and document formatted image p with patches N_p, ColPali / ColQwen2 encodes them as token-level embeddings $E_q = [e_q^1, ..., e_q^{N_q}] \in R^{N_q \times d}$ and patch level $E_p = [e_p^1, ..., e_p^{N_p}] \in R^{N_p \times d}$ in unified embedding space using the LVLM backbone. The relevance score between q and p, denoted as $s(q, p)$, is calculated by firstly identifying the most similar patch embedding in p for each token in q and then summing the similarity scores across all tokens.

$$\begin{aligned} e_p &= \textbf{Retriever}(p) \in R^{N_p \times d} \\ e_q &= \textbf{Retriever}(q) \in R^{N_q \times d} \\ s_j &= \texttt{maxsim}(q_j, p) = \max_i {e_p^i}^T e_q^j \\ s(q, p) &= \sum_j s_j \end{aligned}$$

As illustrated in Figure 2.4, the retrieval process for both page-level and patch-level retrievers can be decomposed to two sequential stages, *i.e.,* offline and online stages. During the **offline** stage, a corpus of documents is collected and pre-encoded into embeddings. During the **online** retrieval stage, the coming queries are also encoded as embeddings. Then the relevance scores between the query embedding and the pre-encoded document embeddings are calculated. Based on these scores, the most relevant documents for each query are retrieved. This two-stage pipeline eliminates the need of repeated page-embedding computations (as they have been

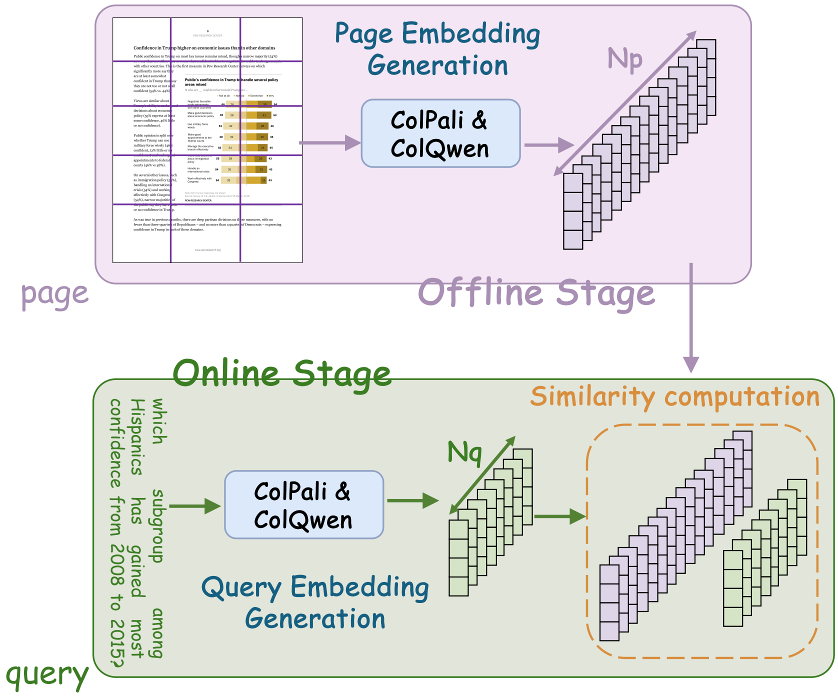


FIGURE 2.4: The retrieval process of VDR. **Top**: Offline stage. **Botton**. Online stage. Patch-level embedding retrievers are adopted in this illustration.

pre-computed during the offline stage). Consequently, it significantly reduces both time and memory costs, and makes the retrieval process more efficient and scalable.

Page-level retrievers are computationally efficient because they minimize the embedding number to only one for each page. However, it comes at the cost of weaker nuanced perceiving and inferior retrieval accuracy. In contrast, patch-level retrievers produce hundreds of embeddings per page (determined by the number of patches divided by the visual encoders). This design provides finer granularity and contributes to superior performance. Nevertheless, these retrievers suffer prohibitive computation demands during both offline indexing and online retrieval. Given the above trade-off, there is an urgent and practical need to develop methods that reduce the number of embeddings in patch-level retrievers without sacrificing their performance. To this end, we review token-reduction approaches as follows.

Token-reduction about LVLM has been extensively explored in recent years. They roughly fall into three categories: (1) *Pruning strategies* [177–180] eliminate low-information tokens based on importance ranking; (2) *Merging strategies* [181–183] combine similar tokens into compressed embeddings; (3) *Hybrid strategies* [184, 185] integrate the above two approaches by preserving high-informative tokens while merging redundant ones. However, previous works primarily focus on token reduction in LVLM generation rather than in retrieval tasks. Different from LVLM generation which focuses on minimizing response latency and FLOPs given specific instructions, token reduction in visualized document retrieval aims to reduce the memory footprint of embeddings. On one hand, it relaxes the constraints on token merging and enables more computationally intensive merging strategies. On the other hand, the absence of queries precludes query-conditioned pruning or merging approaches. Therefore, we draw inspiration from the token reduction in LVLM generation but make targeted modifications to fit the retrieval scenarios.

Lightweight Document Retriever has been explored to address the challenge of large-scale embeddings with two orthogonal approaches: (1) *Dimension Reduction*. ColBERTv2 [186] employs product quantization [187] to reduce the size of each embedding from 768 to 128 dimensions. This design is inherited by ColPali [38] with a simpler projection layer. (2) *Token Reduction*: Clavié et al. [188] introduce the concept of TokenPooling and explore merging strategies for text-based retrievers, *i.e.,* combining similar tokens into one token. A recent blog by ColPali's author [189] further extends this to visualized document retrievers. Following their work, we advance this field by conducting a systematic empirical study on both pruning and merging strategies. Beyond simply proposing a merging strategy, our analysis reveals the limitations of pruning (under retrieval settings) and identifies the optimal combination for merging. Moreover, when compared to the results reported in [189], our fine-tuned Light-ColPali/ColQwen2 presents stronger performance with significantly higher reduction ratios.

Part I

Document Understanding as Information Extraction

Chapter 3

A Unified View on Few-shot Event Detection

We discuss about Information Extraction (as a crucial component of document understanding) from Chapter 3 to Chapter 5. As mentioned in Chapter 1.3, **few-shot** setting is a critical challenge towards IE tasks. In this chapter [1], we focus on conducting a comprehensive empirical study and establishing a unified framework for few-shot event detection. This framework yields a simple yet effective baseline and demonstrates remarkable effectiveness for this IE task.

3.1 Introduction

As introduced in Section 2.1, event detection (ED) is the task of identifying event triggers and types in texts. For example, given *"Cash-strapped Vivendi wants to sell Universal Studios"*, it is to classify the word *"sell"* into a *TransferOwnership* event. The annotation of event instances is costly and labor-consuming, which motivates the research on improving ED with limited labeled samples, *i.e.,* the few-shot ED task. Extensive studies have been carried out on few-shot ED. Nevertheless, there are noticeable discrepancies among existing methods. They adopt different task settings for training and evaluation (*e.g.,* a subset of datasets, annotation

[1] This chapter is published as Yubo Ma, Zehao Wang, Yixin Cao, and Aixin Sun. "Few-shot Event Detection: An Empirical Study and a Unified View". Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11211–11236, Toronto, Canada, 2023. [32]

TABLE 3.1: Noticeable discrepancies among existing few-shot ED methods. **Dataset** indicates the datasets on which the training and/or evaluation is conducted. **Sample Number** refers to the number of labeled samples used. **Sample Source** refers to where training samples come from. Guidelines: example sentences from annotation guidelines. Datasets: subsets of full datasets. Corpus: (unlabeled) external corpus.

	Method	**Experimental setting**		
		Dataset	Sample Number	Sample Source
Prototype-based	Seed-based [190]	ACE	30	Guidelines
	MSEP [191]	ACE	0	Guidelines
	ZSL [82]	ACE	0	Datasets
	DMBPN [79]	FewEvent	{5,10,15}-shot	Datasets
	OntoED [80]	MAVEN / FewEvent	{0,1,5,10,15,20}%	Datasets
	Zhang's [192]	ACE	0	Corpus
	PA-CRF [51]	FewEvent	{5,10}-shot	Datasets
	ProAcT [81]	ACE / FewEvent / RAMS	{5,10}-shot	Datasets
	CausalED [193]	ACE / MAVEN / ERE	5-shot	Datasets
	Yu's [194]	ACE	176	Guidelines + Corpus
	ZED [195]	MAVEN	0	Corpus
	HCL-TAT [85]	FewEvent	{5,10}-shot	Datasets
	KE-PN [84]	ACE / MAVEN / FewEvent	{1,5}-shot	Datasets
Prompt-based	EERC [69]	ACE	{0,1,5,10,20}%	Datasets
	FSQA [70]	ACE	{0,1,3,5,7,9}-shot	Datasets
	EDTE [71]	ACE / ERE	0	-
	Text2Event [73]	ACE / ERE	{1,5,25}%	Datasets
	UIE [74]	ACE / CASIE	{1,5,10}-shot/%	Datasets
	DEGREE [75]	ACE / ERE	{0,1,5,10}-shot	Datasets
	PILED [76]	ACE / MAVEN / FewEvent	{5,10}-shot	Datasets

guidelines, or external corpus) and sample numbers (shot-number or sample-ratio). Table 3.1 provides a detailed comparison of representative methods.

This chapter argues the importance of a unified setting for a better understanding of few-shot ED task. Based on exhaustive background investigation on ED and similar tasks (*e.g.,* NER), we firstly conduct **an empirical study of ten SOTA methods**. We roughly classify the ten methods into two groups: prototype-based models to learn event-type representations and proximity measurement for prediction and prompt-based models that convert ED into a familiar task of Pre-trained Language Models (PLMs).

The second contribution is **a unified view of prototype-based methods** to investigate its superior performance. Instead of picking up the best-performing method as in conventional empirical studies, we take one step further. We break down the design elements along several dimensions, *e.g.,* the source of prototypes, the aggregation form of prototypes, *etc.* And third, through analyzing each effective design element, we propose **a simple yet effective *unified baseline*** that combines all advantageous elements of existing methods. Experiments validate an average 2.7% $F1$ gains. Further analysis in this chapter also provides many valuable insights for future research.

3.2 Empirical Study

3.2.1 Experimental Setting

We detail our comprehensive and fair empirical study as follows.

Dataset source. We utilize three common datasets, *i.e.,* ACE05 [196], MAVEN [197] and ERE [198], to construct few-shot ED datasets in this empirical study. Following the few-shot setting defined in Section 2.1, we downsample sentences from original full training dataset to construct $\mathcal{D}_{train}$ and $\mathcal{D}_{dev}$, and inherit the original test set as the unified $\mathcal{D}_{test}$. For $\mathcal{D}_{train}$ and $\mathcal{D}_{dev}$, we adopt K-shot sampling strategy that each event type has (at least) K samples. Since our sampling is at sentence-level and each sentence could have multiple events, the sampling is NP-complete[2] and unlikely to find a practical solution satisfying exactly K samples for each event type. Therefore, we follow previous work [56, 58] and adopt a greedy sampling algorithm to select sentences, as shown in Algorithm 1. Note that the actual sample number of each event type can be larger than K under this sampling strategy.

Evaluation Metric We use micro-$F1$ score as the evaluation metric. To reduce the random fluctuation, the reported values of each setting are the averaged score and sample standard deviation, of results w.r.t 10 sampled few-shot datasets.

Evaluated methods We evaluate 10 representative methods, including 5 prompt-based and 5 prototype-based methods. To validate the effectiveness of few-shot methods, we also fine-tune a supervised classifier for comparison as a trivial baseline. These approaches are listed as below:

[2]The *Subset Sum Problem*, a classical NP-complete problem, can be reduced to this sampling problem.

Algorithm 1 Greedy Sampling

Require: shot number K, original full dataset $\mathcal{D} = \{(\mathbf{X}, \mathbf{Y})\}$ tagged with label set E

```
Sort $E$ based on their frequencies in $\{\mathbf{Y}\}$ as an ascending order
$S \leftarrow \phi$, Counter $\leftarrow$ dict()
for $y \in E$ do
    Counter$(y) \leftarrow 0$
end for
for $y \in E$ do
    while Counter$(y) < K$ do
        Sample $(\mathbf{X}, \mathbf{Y}) \in \mathcal{D}$ s.t.$\exists j, y_j = y$
        $\mathcal{D} \leftarrow \mathcal{D} \backslash (\mathbf{X}, \mathbf{Y})$
        Update Counter (not only $y$ but all event types in $\mathbf{Y}$)
    end while
end for
for $s \in \mathcal{S}$ do
    $\mathcal{S} \leftarrow \mathcal{S} \backslash s$ and update Counter
    if $\exists y \in E$, s.t. Counter$(y) < K$ then
        $\mathcal{S} \leftarrow \mathcal{S} \bigcup s$
    end if
end for
return $\mathcal{S}$
```

- **Prompt-based** methods leverage the rich language knowledge in PLMs by converting downstream tasks to the task with which PLMs are more familiar.

  (1) *EEQA/EERC* (QA-based) [68, 69]: a QA/MRC-based method which first extracts the trigger word with a natural language query then classifies its type with an additional classifier.

  (2) *EETE* (NLI-based) [71]: a NLI-based method which enumerates all event types and judges whether a clause is entailed by any event.

  (3) *PTE* (cloze task) [199]: a cloze-style prompt method which enumerates each word in the sentence and predicts whether it is the trigger of any event.

  (4) *UIE* (generation-based) [74]: a generation-based method that takes in a sentence and outputs a filled *universal* template, indicating the trigger words and their event types in the sentence.

  (5) *DEGREE* (generation-based) [75]: a generation-based method which enumerates all event types by designing *type-specific* template, and outputs related triggers (if have).

- **Prototype-based** methods predict an event type for each word/span mention by measuring its representation proximity to prototypes.

  (6) *ProtoNet* [52], (7) *L-TapNet-CDT* [54], (8) *PA-CRF* [51], (9) *CONTAINER* [57], (10) *FSLS* [58]. See Table 3.3 for more details.

Implementation Details We unify PLMs in each method as much as possible for a fair comparison in our empirical study. Specifically, we use `RoBERTa-base` [200] for all prototype-based methods and three non-generation prompt-based methods. However, we keep the method's original PLM for two prompt-based methods with generation prompt, UIE (`T5-base`) [43] and DEGREE (`BART-large`) [42]. We observe their performance collapses with smaller PLMs. For all methods, we initialize their pre-trained weights and further train them using Huggingface library.[3] Each experiment is run on single NVIDIA-V100 GPU, and the final reported performance for each setting (*e.g.,* ACE 2-shot) is the averaged result w.r.t ten distinct few-shot training datasets which are sampled with different random seeds. We further detail the implementation of all methods.

- **Prompt-based methods** We keep all other hyperparameters the same as in their original papers, except learning rates and epochs. We grid-search best learning rates in [1e-5, 2e-5, 5e-5, 1e-4] for each setting. As for epochs, we find the range of appropriate epochsis highly affected by the prompt format. Therefore we search for epochs method by method without a unified range.

- **Prototype-base methods** We build a codebase based on the unified view. We then implement these methods directly on the unified framework, by having different choices for each design element. To ensure the correctness of our codebase, we also compare between results obtained from our implementation and original code for each method, and find they achieving similar performance on few-shot ED datasets. For all methods (including *unified baseline*), we train them with the AdamW optimizer with linear scheduler and 0.1 warmup step. We set weight-decay coefficient as 1e-5 and maximum gradient norms as 1.0. We add a 128-long window centering on the trigger words and only encode the words within the window; in other words, the maximum encoding sequence length is 128. The batch size is set as 128, and

[3] https://huggingface.co/

TABLE 3.2: Overall results of *fine-tuning* method, 10 existing few-shot ED methods, and the *unified baseline*. The best results are in bold face and the second best are underlined. The results are averaged over 10 repeated experiments, and sample standard deviations are in the round bracket. The standard deviations are derived from different **sampling few-shot datasets** instead of **random seeds**. Thus high standard deviation values do not mean that no significant difference among these methods.

	Method	ACE05			MAVEN			ERE		
		2-shot	5-shot	10-shot	2-shot	5-shot	10-shot	2-shot	5-shot	10-shot
	Fine-tuning	33.3 (4.4)	42.5 (4.6)	48.2 (1.5)	40.8 (4.7)	52.1 (0.7)	55.7 (0.2)	32.9 (2.1)	39.8 (2.9)	43.6 (1.7)
Prompt-based	EEQA/EERC	24.1 (12.2)	43.1 (2.7)	48.3 (2.4)	33.4 (9.2)	48.1 (0.9)	52.5 (0.5)	13.7 (8.6)	34.4 (1.7)	39.8 (2.4)
	EETE	15.7 (0.6)	19.1 (0.3)	21.4 (0.2)	28.9 (4.3)	30.6 (1.3)	32.5 (1.1)	10.6 (2.3)	12.8 (2.2)	13.7 (2.8)
	PTE	38.4 (4.2)	42.6 (7.2)	49.8 (1.9)	41.3 (1.4)	46.0 (0.6)	49.5 (0.6)	33.4 (2.8)	36.9 (1.3)	37.0 (1.8)
	UIE	29.3 (2.9)	38.3 (4.2)	43.4 (3.5)	33.7 (1.4)	44.4 (0.3)	50.5 (0.5)	19.7 (1.5)	30.8 (1.9)	34.1 (1.6)
	DEGREE	40.0 (2.9)	45.5 (3.2)	48.5 (2.1)	43.3 (1.0)	43.4 (5.9)	45.5 (4.3)	31.3 (3.1)	36.0 (4.6)	40.7 (2.2)
Prototype-bsd	ProtoNet	38.3 (5.0)	47.2 (3.9)	52.3 (2.4)	44.5 (2.2)	51.7 (0.6)	55.4 (0.2)	31.6 (2.7)	39.7 (2.4)	44.3 (2.3)
	PA-CRF	34.9 (7.2)	48.1 (3.9)	51.7 (2.6)	44.8 (2.2)	51.8 (1.0)	55.3 (0.4)	30.6 (2.8)	38.0 (3.9)	40.4 (2.0)
	L-TapNet-CDT	43.2 (3.8)	49.8 (2.9)	53.5 (3.4)	48.6 (1.2)	53.2 (0.4)	56.1 (0.9)	35.6 (2.6)	42.7 (1.7)	45.1 (3.2)
	CONTAINER	40.1 (3.8)	47.7 (3.3)	50.1 (1.8)	44.2 (1.4)	50.8 (0.9)	52.9 (0.3)	34.4 (3.6)	39.3 (1.9)	44.5 (2.3)
	FSLS	39.2 (3.4)	47.5 (3.2)	51.9 (1.7)	46.7 (1.2)	51.5 (0.5)	56.2 (0.2)	34.5 (3.1)	39.8 (2.5)	44.0 (2.0)
	Unified Baseline	**46.0** (4.6)	**54.4** (2.6)	**56.7** (1.5)	**49.5** (1.7)	**54.7** (0.8)	**57.8** (1.2)	**38.8** (2.4)	**45.5** (2.8)	**48.4** (2.6)

training steps as 200 if the transfer function is scaled otherwise 500. We grid-search best learning rates in [1e-5, 2e-5, 5e-5, 1e-4] for each setting. For ProtoNet and its variants, we further split the sentences into support set and query set. The number in support set K_S and query set K_Q are (1, 1) for 2-shot settings, (2, 3) for 5-shot settings. The split strategy is (2, 8) for 10-shot dataset constructed from MAVEN and (5, 5) for others. For methods adopting MoCo-CL setting , we maintain a queue storing sample representations with length 2048 for ACE/ERE 2-shot settings and 8192 for others. For methods adopting CRF, we follow default hyperparameters about CRF in their original papers. For methods adopting scaled transfer functions, we grid search the scaled coefficient τ in [0.1, 0.2, 0.3].

3.2.2 Results

We show the results of the 10 methods under the low-resource setting in Table 3.2 and discuss as follows.

Fine-tuning. Despite its simpleness, fine-tuning achieves acceptable performance. In particular, it is even comparable to the strongest existing methods on MAVEN dataset (only being 1.1% and 0.5% less under 5-shot and 10-shot settings). One possible reason that fine-tuning is good on MAVEN is that MAVEN has 168 event types, much larger than others. When the absolute number of samples is relatively large, PLMs might capture implicit interactions among different event types, even though the samples per event type are limited. When the sample number is scarce, however, fine-tuning is much poorer than existing competitive methods (see ACE05). Thus, we validate the necessity and progress of existing few-shot methods.

Prompt-based methods. Prompt-based methods deliver much poorer results than expected, even compared to fine-tuning, especially when the sample number is extremely scarce. It shows designing effective prompts for ED tasks with very limited annotations is still challenging. We speculate it is due to the natural gap between ED tasks (sequence labeling or span extraction) and pre-training tasks in PLMs (sentence classification or generation).

Among prompt-based methods, PTE and DEGREE achieve relatively robust performance under all settings. DEGREE is advantageous when the sample size is small, but it cannot well handle a dataset with many event types like MAVEN. Note that, DEGREE enumerates event types to query their potential triggers; both efficiency and effectiveness drop with the increasing number of event types. When sample sizes are relatively large, EEQA shows competitive performance as well.

3.3 A Prototype-based Unified View

Due to the superior performance shown in last section, we zoom into prototype-based methods to provide a unified view towards a better understanding. We observe that they share lots of similar components. As shown in Table 3.3 and Figure 3.1, we decompose prototype-based methods into 5 design elements: prototype source, transfer function, distance function, aggregation form, and CRF module. This unified view enables us to compare choices in each design element directly. By aggregating the effective choices, we end with a *Unified Baseline*.

Formally, given a dataset $\mathcal{D}$ annotated with schema E (the set of event types) and a sentence $X = [x_1, ..., x_N]^T \in \mathcal{D}$, where x_i is the i-th word and N the length of this

TABLE 3.3: Decomposing five prototype-based methods and *unified baseline* along design elements. "Both" in column 1 means both event mentions and label names for y are prototype sources. JSD: Jensen–Shannon divergence. $\mathcal{M}$: Projection matrix in TapNet. $\mathcal{N}(\mu(h), \Sigma(h))$: Gaussian distribution with mean $\mu(h)$and covariance matrix $\Sigma(h)$.

Method	Prototype $\mathcal{C}_y$	Aggregation	Distance $d(u, v)$	Transfer $f(h)$	CRF Module
ProtoNet [52]	Event mentions	feature	$\|\|u-v\|\|_2$	h	–
L-TapNet-CDT [54]	Both	feature	$-u^T v/\tau$	$\mathcal{M}\frac{h}{\|\|h\|\|}$	CRF-Inference
PA-CRF [51]	Event mentions	feature	$-u^T v$	$\frac{h}{\|\|h\|\|}$	CRF-PA
CONTAINER [57]	Event mentions	score	JSD$(u\|\|v)$	$\mathcal{N}(\mu(h), \Sigma(h))$	CRF-Inference
FSLS [58]	Label name	–	$-u^T v$	h	–
Unified Baseline (Ours)	Both	score + loss	$-u^T v/\tau$	$\frac{h}{\|\|h\|\|}$	–

sentence, ED aims to assign a label $y_i \in (E \cup \{\texttt{N.A.}\})$ for each x_i in X. We say that word x_i triggering an event y_i if $y_i \in E$. Unless otherwise stated, we ignore the subscript i for convenience in later description.

In prototype-based methods, we define P-score(x, y) to measure the proximity between an event mention x and event type y, and predict the likelihood $p(y|x)$ from P-score(x, y) for each $y \in (E \cup \{\texttt{N.A.}\})$

$$p(y|x) = \text{Softmax}_{y\sim(E\cup\{\texttt{N.A.}\})}\text{P-score}(x, y)$$

The general framework is as follows. Denote the PLM's output representation of event mention x and data c_y in <u>prototype source</u> $\mathcal{C}_y$ as h_x and h_{c_y} respectively, where $h \in R^m$ and m is the dimension of PLM's hidden space. The first step is to convert h_x and h_{c_y} to appropriate representations via a <u>transfer function</u> $f(\cdot)$. Then the methods maintain either a single or multiple prototypes c_y's for each event type, determined by the adopted <u>aggregation form</u>. Third, the distance between $f(h_x)$ and $f(h_{c_y})$ (single prototype) or $f(h_{c_y})$'s (multiple prototypes) is computed via a <u>distance function</u> $d(\cdot, \cdot)$ to learn the proximity scores P-score(x, y). Finally, an optional <u>CRF module</u> is used to adjust P-score(x, y) for x in the same sentence to model their label dependencies. For inference, we adopt nearest neighbor classification by assigning the sample with nearest event type in $\cup_{y\in(E\cup\{\texttt{N.A.}\})}\mathcal{C}_y$

$$\hat{y}_x = \underset{y\in(E\cup\{\texttt{N.A.}\})}{\text{argmin}} \min_{c_y\in\mathcal{C}_y} d(f(h_x), f(h_{c_y}))$$

Next, we detail the five design elements:

Prototype source $\mathcal{C}_y$ (purple circles in Figure 3.1, same below) indicates a set about the source of data / information for constructing the prototypes. There are mainly two types of sources:

- **Event mentions** (purple circle without words): ProtoNet and its variants in Figure 3.1(b),(c),(d) additionally split a support set $\mathcal{S}_y$ from training data as prototype source, while contrastive learning methods in Figure 3.1(a) view every annotated mention as the source (except the query one).
- **Label semantics** (purple ellipses with words): Sometimes, the label name l_y is utilized as the source to enhance or directly construct the prototypes. For example, FSLS in Figure 3.1(e) views the text representation of type names as prototypes, while L-TapNet-CDT in Figure 3.1(c) utilizes both the above kinds of prototype sources.

Transfer function $f : R^m \to R^n$ (yellow modules) transfers PLM outputs into the distance space for prototype proximity measurement. Widely used transfer functions include normalization in Figure 3.1(b), down-projection in Figure 3.1(c), reparameterization in Figure 3.1(a), or an identity function.

Distance function $d : R^n \times R^n \to R_+$ (green modules) measures the distance of two transferred representations within the same embedded space. Common distance functions are euclidean distance in Figure 3.1(d) and negative cosine similarity in Figure 3.1(b),(c),(e).

Aggregation form (blue modules) describes how to compute P-score(x, y) based on a single or multiple prototype sources. Aggregation may happen at three levels.

- **Feature-level**: ProtoNet and its variants in Figure 3.1(b),(c),(d) aims to construct a *single* prototype $h_{\bar{c}_y}$ for each event type y by merging various features, which ease the calculation P-score$(x, y) = -d(f(h_x), f(h_{\bar{c}_y}))$.
- **Score-level**: CONTAINER in Figure 3.1(a) views each data as a prototype (they have *multiple* prototypes for each type y) and computes the distance $d(f(h_x), f(h_{c_y}))$ for each $c_y \in \mathcal{C}_y$. These distances are then merged to obtain P-score(x, y).

- **Loss-level**: Such form has multiple parallel branches b for each mention x. Each branch has its own P-score$^{(b)}(x, y)$ and is optimized with different loss components during training. Thus it could be viewed as a multi-task learning format. See *unified baseline* in Figure 3.1(f).

CRF module (orange modules) adjusts predictions within the same sentence by explicitly considering the label dependencies between sequential inputs. The vanilla CRF [201] and its variants in Figure 3.1(a),(b),(c) post additional constraints into few-shot learning.

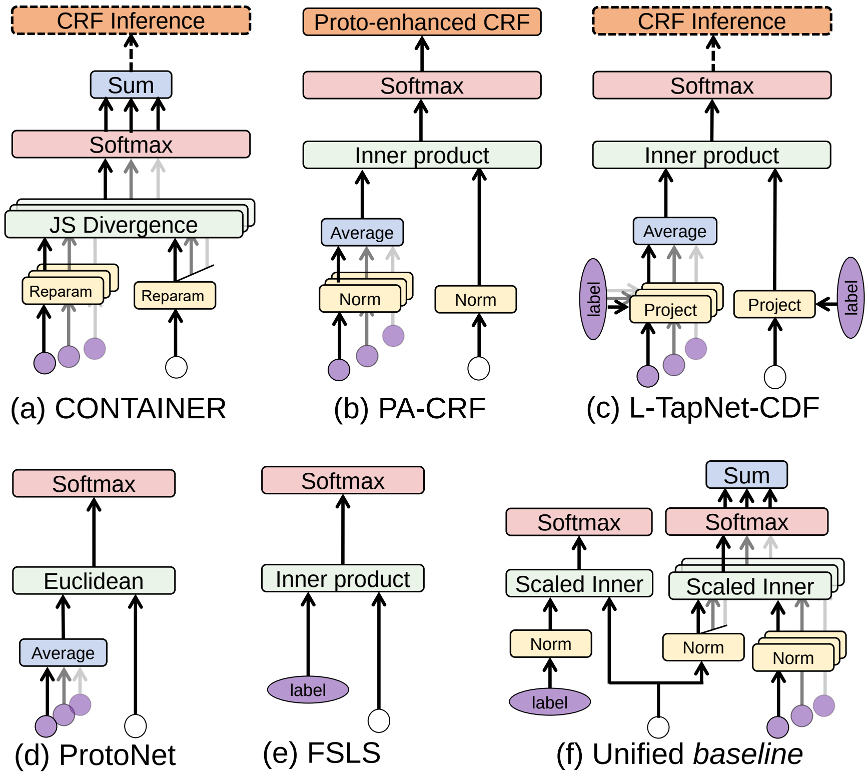


FIGURE 3.1: The architectures of five existing prototype-based methods and the unified baseline. Given event mention x and event type y, each sub-figure depicts how to compute the P-score(x, y). White circles: representation of predicted event h_x. Purple circles: representation of prototypes h_{c_y} ($c_y \in \mathcal{C}_y$). Yellow modules: transfer functions. Green modules: distance functions. Blue modules: aggregation form. Orange modules: CRF modules. Dashed lines in (a) and (c) represent that their CRFs are only used during inference.

TABLE 3.4: Variants on distance function $d(u, v)$ (top) and transfer function $f(h)$ (bottom).

Distance function	$d(u, v)$
Cosine similarity (S)	$u^T v$
Scaled cosine similarity (SS)	$u^T v/\tau$
JS Divergence (KL)	$\text{JSD}(u\|\|v)$
Euclidean distance (EU)	$-\|\|u - v\|\|_2$
Scaled euclidean distance (SEU)	$-\|\|u - v\|\|_2/\tau$
Transfer function	$f(h)$
Identify (I)	h
Down-projection (D)	$\mathcal{M}h$
Reparameterization (R)	$\mathcal{N}(\mu(h), \Sigma(h))$
Normalization (N)	$h/\|\|h\|\|$
Down-projection + Normalization (DN)	$\mathcal{M}h/\|\|h\|\|$

3.4 Towards a Unified baseline

Since prototype-based methods have overall better results, we zoom into the design elements to search for effective choices based on the unified view.

3.4.1 Transfer function and Distance function

We consider several combinations about distance and transfer functions listed in Table 3.4. We choose cosine similarity (S), negative euclidean distance (EU) and their scaled version (SS/SEU) as distance functions. And we pick out identify (I), down-projection (D) and their normalization version (N/DN) as transfer function. We additionally consider the KL-reparameterization combination (KL-R) used in CONTAINER.

We conduct experiments with four existing prototype-based methods[4] by only changing their transfer and distance functions. We illustrate their results on ACE dataset in Figure 3.2. (1) From comparison about performance in ProtoNet and TapNet, we find TapNet, *i.e.,* the down-projection transfer, shows no significant improvement on few-shot ED tasks. (2) A scaled coefficient in distance function

[4]We *degrade* L-TapNet-CDT to TapNet, and do not include PA-CRF here, because CRF and label-enhancement are not the factors considered in this subsection.

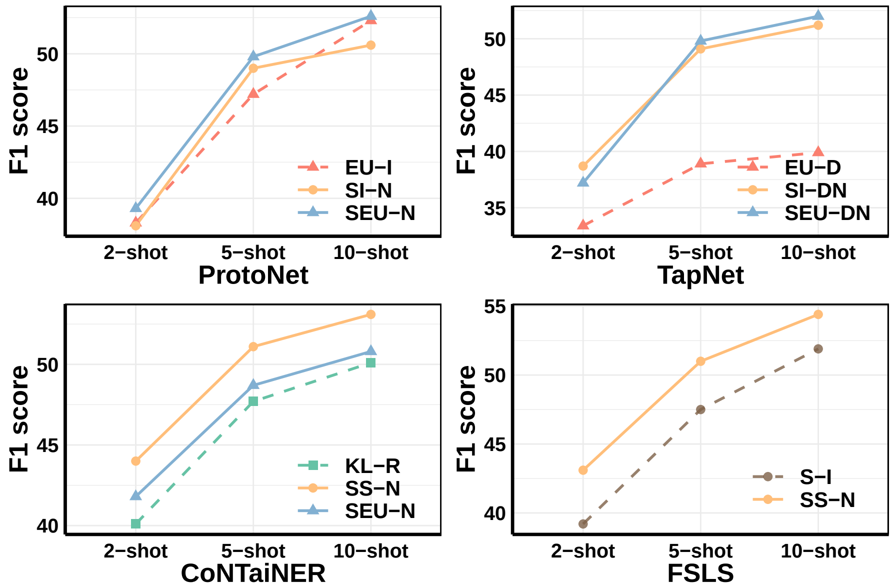


FIGURE 3.2: Performance of different (d, f) combinations on ACE05.

achieves strong performance with normalization transfer function, while the performance collapses (failing to converge) without normalization. (3) For ProtoNet and TapNet, scaled euclidean distance (SEU) is a better choice for distance function, while other methods prefer scaled cosine similarity (SS). Based on the findings above, we substitute d and f to the most appropriate for all existing methods and observe a significant improvement on all three datasets, as shown in Table 3.5.

TABLE 3.5: Performance comparison of methods w/ and w/o adjustment on distance function d and transfer function f. The most appropriate distance functions are scaled euclidean distance (SEU) for ProtoNet and TapNet and scaled cosine similarity (SS) for other two. The most appropriate transfer function is normalization (N) for all four existing methods. The results are averaged among 10 repeated experiments and sample standard deviations are in round brackets. We highlight the better one for each method *w/* and *w/o* adjustment.

Methods		ACE05			MAVEN			ERE		
		2-shot	5-shot	10-shot	2-shot	5-shot	10-shot	2-shot	5-shot	10-shot
ProtoNet	w/o adjust	38.3(5.0)	47.2(3.9)	52.3(2.4)	44.5(2.2)	51.7(0.6)	55.4(0.2)	31.6(2.7)	39.7(2.4)	**44.3**(2.3)
	w/ adjust	**39.3**(4.6)	**49.8**(4.3)	**52.6**(1.9)	**46.7**(1.6)	**52.8**(0.6)	**56.5**(0.6)	**32.6**(3.0)	**40.1**(1.9)	44.2(1.9)
TapNet	w/o adjust	**38.7**(4.3)	49.1(4.5)	51.2(1.7)	45.7(1.8)	51.7(1.1)	55.0(0.7)	35.3(3.8)	40.2(2.5)	44.7(2.9)
	w/ adjust	37.2(5.6)	**49.8**(3.1)	**52.0**(1.9)	**46.1**(1.9)	**51.9**(0.6)	55.0(0.6)	**37.0**(4.0)	**43.4**(1.9)	**46.4**(2.9)
CONTAINER	w/o adjust	40.1(3.8)	47.7(3.3)	50.1(1.8)	44.2(1.4)	50.8(0.9)	52.9(0.3)	34.4(3.6)	39.3(1.9)	44.5(2.3)
	w/ adjust	**44.0**(3.2)	**51.1**(1.1)	**53.1**(1.8)	**44.6**(1.7)	**52.1**(0.5)	**55.1**(0.4)	**36.5**(4.1)	**42.0**(1.9)	**45.4**(1.5)
FSLS	w/o adjust	39.2(3.4)	47.5(3.2)	51.9(1.7)	46.7(1.2)	51.5(0.5)	**56.2**(0.2)	34.5(3.1)	39.8(2.5)	44.0(2.0)
	w/ adjust	**43.1**(3.4)	**51.0**(2.4)	**54.4**(1.5)	**48.3**(1.6)	**53.4**(1.6)	56.1(0.7)	**35.7**(2.1)	**40.6**(2.4)	**45.4**(1.7)

3.4.2 CRF module

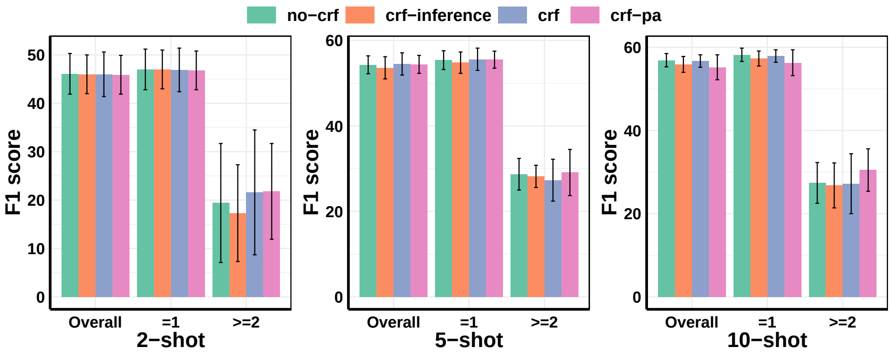
We explore whether CRF improves the performance of few-shot ED task. Based on *Ll-MoCo* model, we conduct experiment with three different CRF variants, CDT (CRF inference [54]), vanilla CRF [201] and PA-CRF [51], on ACE05 and MAVEN datasets. Their results are in Figure 3.3. It shows different CRF variants achieve similar result compared with model without CRF, while a trained CRF (and its prototype-enhanced variant) slightly benefits multiple-word triggers when the sample is extremely scarce (see ACE05 2-shot). These results are inconsistent with other similar sequence labeling tasks such as NER or slot tagging, in which CRF usually significantly improves model performance. We speculate it is due to that the pattern of triggers in ED task is relatively simple. To validate such assumption, we count all triggers in ACE05 and MAVEN datasets. We find that above 96% of triggers are single words, and most of the remaining triggers are verb phrases (only about 0.5% of triggers are phrases having three or more words with complicated structure). Thus the explicit modeling of transfer dependency among different event types is somewhat not very meaningful under few-shot ED task. Hence, we drop CRF module in the *unified baseline*.

3.4.3 Prototype source

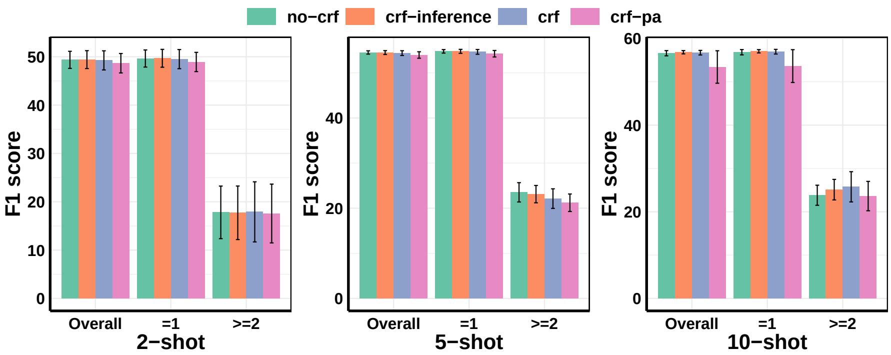
We explore whether label semantic and event mentions are complementary prototype sources, *i.e.,* whether utilizing both achieves better performance than either one. We choose ProtoNet and FSLS as base models which contain only a single kind of prototype source (mentions or labels). Then we combine the two models using three aggregating forms mentioned in Section 3.3 and list full results on all three datasets in Table 3.6. The results further validate our claims: (1) leveraging both label semantics and mentions as prototype sources improve performance under almost all settings. (2) Merging the two kinds of sources at the loss-level is the best choice among the three aggregation alternatives.

3.4.4 Contrastive Learning

Contrastive Learning (CL) [86] is initially developed for self-supervised representation learning and is recently used to facilitate supervised learning as well. It pulls



(A) ACE05



(B) MAVEN

FIGURE 3.3: Overall performance of different CRF variants on ACE05 and MAVEN datasets. We also provide performance grouped by trigger word length: $=1$: single trigger words. ≥ 2: trigger phrases.

TABLE 3.6: Performance with different (1) prototype sources and (2) aggregation form. **ProtoNet**: only event mentions. **FSLS**: label semantic. **Lf-ProtoNet**: aggregate two types of prototype sources at feature-level. **Ls-ProtoNet**: at score-level. **Ll-ProtoNet**: at loss-level. The results are averaged over 10 repeated experiments and sample standard deviations are in round brackets.

Methods	ACE05			MAVEN			ERE		
	2-shot	5-shot	10-shot	2-shot	5-shot	10-shot	2-shot	5-shot	10-shot
ProtoNet	39.3(4.6)	49.8(4.3)	52.6(1.9)	46.7(1.6)	52.8(0.6)	56.0(0.6)	32.6(3.0)	40.1(1.9)	44.2(1.9)
FSLS	<u>43.0</u>(3.4)	50.6(2.4)	**54.1**(1.5)	48.3(1.6)	53.4(0.2)	56.1(0.7)	35.7(2.1)	40.6(2.4)	**45.4**(1.7)
Lf-ProtoNet	41.9(3.8)	50.8(3.0)	52.9(2.4)	49.0(1.1)	53.4(1.0)	56.3(0.7)	35.3(3.6)	<u>41.8</u>(1.8)	45.3(2.2)
Ls-ProtoNet	42.7(4.8)	**51.2**(2.9)	52.7(1.7)	<u>49.3</u>(1.9)	<u>53.5</u>(0.7)	<u>56.5</u>(0.1)	<u>36.0</u>(2.5)	41.3(3.6)	44.8(2.5)
Ll-ProtoNet	**43.3**(4.0)	<u>50.9</u>(2.7)	<u>53.0</u>(2.1)	**50.2**(1.5)	**54.3**(0.8)	**56.7**(0.6)	**37.6**(3.1)	**43.0**(2.4)	<u>45.3</u>(1.9)

samples with same labels together while pushes samples with distinct labels apart in their embedding space. We view CL as a *generalized* format of prototype-based methods and include it to the unified view. Under such view, every sample is a prototype and each single event type could have multiple prototypes. Given an event mention, its distances to the prototypes are computed and aggregated by event types to determine the overall distance to each event type.

Two types of Contrastive Learning We name the **representation** of event mention as query and prototypes (*i.e.,* other event mentions) as keys. Then CL could be further split into two cases, in-batch CL [202] and MoCo CL [203], according to where their **keys** are from. In-batch CL views other event mentions within the same batch as the keys, and the encoder for computing the queries and keys in batch-CL is updated end-to-end by back-propagation. For MoCo CL, the encoder for key is momentum-updated along the encoder for query, and it accordingly maintains a queue to store keys and utilizes them multiple times once they are previously computed. We refer readers to MoCo CL [203] for the details of in-batch CL and MoCo CL.

CONTAINER [57] adopts in-batch CL setting for few-shot NER model and we transfer it to ED domain in our empirical study. We further compare the two types of CL for our *unified baseline* with effective components and present the full results in Table 3.4.4. We observe in-batch CL outperforms MoCo-CL when the number of the sentence is small, and the situation reverses with the increasing of sentence number. We speculate it is due to two main reasons: (1) When all sentences could be within the single batch, in-batch CL is a better approach since it computes and updates all representations of keys and queries end-to-end by back propagation, while MoCo-CL computes the key representation by a momentum-updated encoder with gradient stopping. When the sentence number is larger than batch size, however, in-batch CL lose the information of some samples in each step, while MoCo-CL keeps all samples within the queue and leverages these approximate representations for a more extensive comparison and learning. (2) MoCo-CL also has an effect of data-augmentation under few-shot ED task, since the sentence number is usually much smaller than the queue size. Then the queue would store multiple representations for each sample, which are computed and stored in different previous steps. The benefits of such data augmentation take effect when there are relatively abundant sentences and accordingly diverse augmentations.

3.4.5 The unified baseline

Here is a summary of the findings: (1) Scaled euclidean or cosine similarity as distance measure with normalized transfer benefits existing methods. (2) CRF modules show no improvement in performance. (3) Label semantic and event mentions are complementary prototype sources, and aggregating them at loss-level is the best choice. (4) As for the branch of event mentions, CL is more advantageous than PL for few-shot ED tasks. (5) MoCo CL performs better when there are a good number of sentences, otherwise in-batch CL is better.

Based on these findings, we develop a simple but effective *unified baseline* as follows. We utilize both label semantic and event mentions as prototype sources and aggregate two types of sources at loss-level. Specifically, we assign two branches with their own losses for label semantic and event mentions respectively. Both two branches adopt scaled cosine similarity $d_\tau(u, v) = -\frac{u^T v}{\tau}$ as distance measure and normalization $f(h) = h/\|h\|_2$ as transfer function. We do not add CRF modules.

For label semantic branch, we follow FSLS and set the embeddings of event name as prototypes. Here h_x and h_{e_y} represent the PLM representation of event mention x and label name e_y, respectively.

$$e_y = \text{Event\_name}(y)$$
$$\text{logits}^{(l)}(y|x) = -d_\tau(f(h_x), f(h_{e_y}))$$

For event mention branch, we adopt CL which aggregates prototype sources (event mentions) at score-level. If the total sentence number in train set is smaller than 128, we take in-batch CL (CONTAINER) strategy as below:

$$\text{logits}^{(m)}(y|x) = \sum_{x' \in \mathcal{S}_y(x)} \frac{-d(f(h_x), f(h_{x'}))}{|\mathcal{S}_y(x)|}$$

$\mathcal{S}_y(x) = \{x'|(x', y') \in D, y' = y, x' \neq x\}$ is the set of all other mentions with the same label. If the total sentence number in train set is larger than 128, we instead take MoCo CL maintaining a queue for $\mathcal{S}_y(x)$ and a momentum encoder.

We then calculate the losses of these two branches and merge them for joint optimization:

$$p^{(l/m)}(y|x) = \text{Softmax}_y[\text{logits}^{(l/m)}(y|x)]$$
$$L^{(l/m)}(y|x) = -\sum_{(x,y)} y\log(p^{(l/m)}(y|x))$$
$$L = L^{(l)} + L^{(m)}$$

The diagram of the *unified baseline* is illustrated in Figure 3.1(f) and its performance is shown in Table 3.2. Clearly, *unified baseline* outperforms all existing methods significantly, 2.7% $F1$ gains on average, under all few-shot settings.

3.5 Conclusion

We have conducted a comprehensive empirical study comparing 12 representative methods under few-shot event detection setting. For systematic analysis, we proposed a unified framework of promising prototype-based methods. Based on it, we present a simple and effective *baseline* that outperforms all existing methods significantly under *few-shot* setting. In the future, we aim to explore how to leverage unlabeled corpus for few-shot ED tasks, such as data augmentation, weakly-supervised learning, and self-training.

Chapter 4

Prompting Argument Interaction for Event Argument Extraction

In the last chapter, we tackle the **few-shot** challenge in event detection (ED) task. And in this chapter [1], we turn attention to its subsequent task: Event Argument Extraction (EAE). We aim to simultaneously address two key challenges, *i.e.,* **few-shot** and **large-scale**, in EAE task as follows.

4.1 Introduction

Understanding text by identifying the event and arguments has been a long-standing goal [204]. For example, given the document snippet,

> *Cash-strapped Vivendi wants to sell Universal Studios, its Universal theme parks and television production company.*

it is talking about a *Sell* event, with four involved arguments, *i.e., Vivendi* (Seller), *Universal Studios* (Artifact), *parks* (Artifact), and *company* (Artifact), where the role schemata are in brackets and the corresponding arguments are in italic. Since

[1] This chapter is published as Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. "Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction". In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6759-6774, Dublin, Ireland, 2022. [33]

we have addressed the few-shot ED task in the last chapter, the main challenge lies in Event Argument Extraction (EAE).

In EAE task, each event type contains multiple specific roles. This schema structure creates two major challenges mentioned in Section 1.3. **(1) Few-shot**: The fine-grained role definitions require extensive annotation efforts, and make data collection time-consuming and expensive. **(2) Large-scale**: The sheer volume of arguments that need to be extracted poses significant processing demands in short time. To address these challenges, this chapter focuses on developing an effective and efficient EAE system. And we specifically examine the following key questions:

- How to extract all arguments simultaneously for efficiency?
- How to effectively capture argument interactions for long text, without knowing them in advance?
- How to elicit more knowledge from PLMs to lower the needs of annotation?

To this end, we investigates prompt tuning under an extractive setting and proposes a novel method **PAIE** that **P**rompting **A**rgument **I**nteractions for **E**AE. It extends QA-based models to handle multiple argument extraction and meanwhile takes the best advantage of PLMs. The basic idea is to design suitable templates to prompt all argument roles for PLMs, and obtain role-specific queries to jointly select optimal spans from the text. Thus, instead of unavailable arguments, each role in the template serves as a slot for interactions, and during learning, PLMs tend to fill these slots with exact arguments via a matching loss. By predicting arguments together, PAIE enjoys an efficient and effective learning procedure. Besides, the inter-event knowledge transfer between similar role prompts alleviates the heavy burden of annotation cost.

Specifically, for prompting extraction, this chapter designs two span selectors based on role prompts, which select start/end tokens among input texts. This chapter explores three types of prompts: manual template, concatenation template, and soft prompt. They perform well at both sentence-level EAE (S-EAE) and document-level EAE (D-EAE) and ease the requirements of the exhaustive prompt design. For joint span selection, this chapter designs a bipartite matching loss that makes the least-cost match between predictions and ground truth so that each argument

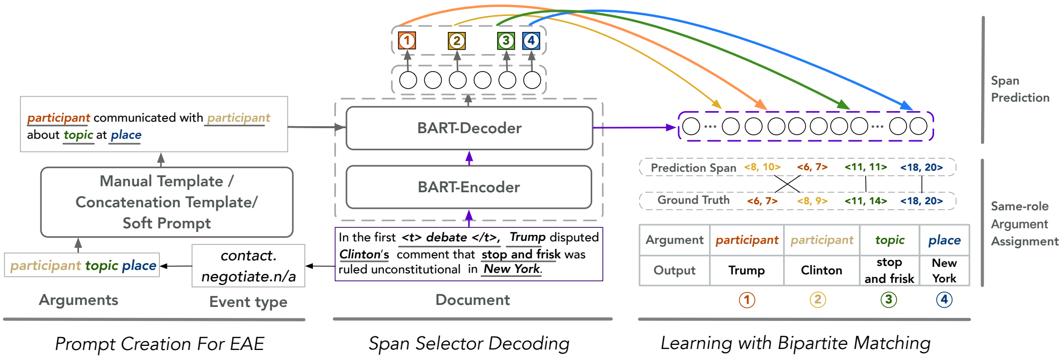


FIGURE 4.1: The overall architecture of PAIE. Given a context (about an event), PAIE first creates joint prompts based on its event type. Then the context and prompt are fed into the BART-Encoder and BART-Decoder to generate context representation and role-specific span selectors. Multiple span selectors extract argument spans from the context simultaneously. A bipartite matching loss finally optimizes the global span assignment.

will find the optimal role prompt. It can also deal with multiple arguments with the same role via flexible role prompts instead of heuristic threshold tuning. The contribution of this chapter is summarized as follow:

- Propose a novel model, PAIE, that is effective and efficient for S-EAE and D-EAE, and robust to the few-shot setting.
- Formulate and investigate prompt tuning under extractive settings, with a joint selection scheme for optimal span assignments.
- Conduct extensive experiments on three benchmarks. The results show a promising improvement with PAIE (3.5% and 2.3% F1 gains on average absolutely in base and large model). Further ablation study demonstrates the efficiency under large-scale scenarios and generalization to few-shot settings of our proposed model, as well as the effectiveness of prompt tuning for extraction.

4.2 Methodology

PAIE considers multiple arguments and their interactions to prompt PLMs for joint extraction. Our model, as illustrated in Figure 4.1, contains three core components: *prompt creation*, *span selector decoding*, and *span prediction*. In the following sections, we will first formulate the problem and describe each component in turn.

4.2.1 Formulating Prompt for Extraction

Existing prompt-based methods mainly focus on classification and generation tasks. Conventional extraction objectives are converted into a generation task. This brings an inefficiency issue that the model has to enumerate all of extraction candidates. For example, Cui *et al.* [205] design the prompt for named entity recognition: *[candidate span] is [entity type/not a] entity*. The models need to fill the first slot with candidate entities, and check the outputs of LM for the second slot for extraction. Can prompt-based methods directly be applied on extraction? We give a formulation about the general extractive prompting method and then apply it on EAE for a case study.

- **Prompt Creation**. Given context X and a series of queries $Q = \{q_1, q_2, ..., q_K\}$, we create a joint prompt containing all these queries, where f_{prompt} is the prompt creator.

$$Pt = f_{prompt}(Q)$$

- **Prompted Selector Decoding**. Given a PLM $\mathcal{L}$, context X, and prompt Pt, we decode a query-specific (answering) span selector as follows:

$$\theta_{q_k} = h_{\mathcal{L}}(q_k; Pt, X)$$

  where q_k is the k-th query in the prompt and $h_{\mathcal{L}}$ is the outputs of PLMs.

- **Prompted Span Selection**. To find the optimal span, we design two selectors for the start and end tokens from context:

$$(s, e)_{q_k} = \text{Span-search}[g_{\mathcal{L}}(X; \theta_q)]$$

  where $(s, e)_{q_k}$ is the span about k-th query and $g_{\mathcal{L}}$ is the span selector. Clearly, such formulation is better than generative extraction by mainly considering the adjacent constraints of span.

4.2.2 Prompt Creation for EAE

We create a set of prompts for each event type e in dataset D. Each prompt contains all roles $r \in R^{(e)}$. For example in Figure4.1, given event type e as *negotiate* and $R^{(e)}$ as $\{Participant, Topic, Place\}$, the prompt $Pt^{(e)}$ may be defined as follows:

Participant communicated with Participant about Topic at Place .

We call the mentions of roles in the prompt as **slot**, and there are four slots underlined in this example (and colored in Figure 4.1). Such design allows our model to capture the implicit interactions among different roles.

To avoid threshold tuning for multiple arguments with the same role, the prompt is flexible to use multiple slots for the same role, such as role *Participant* in the above example. The number of slots for the role is heuristically determined according to the maximum number of arguments of each role in the training dataset. We design three different prompt creators f_{prompt}, the mapping from a set of roles to a prompt as follows:

- **Manual Template**: All roles are connected manually with natural language. We follow the template from BART-Gen [89] for fair comparison.
- **Soft Prompt**: Following SoftPrompt [206] and P-tuning [207], we connect different roles with learnable, role-specific pseudo tokens.
- **Concatenation Template**: To concatenate all role names belonging to one event type.

We give one example of these three types of prompt in Table 4.1.

4.2.3 Role-specific Selector Generation

Given context X and prompt Pt, this module generates the role-specific span selector θ_k, for each slot k of the prompt. Here we choose $\mathcal{L}$ as BART [42], a standard Transformer-based pre-trained language model consisting both an **Encoder** and a **Decoder**: $\mathcal{L} = [\mathcal{L}_{enc}, \mathcal{L}_{dec}]$.

Prompt Type	**Prompt Example**
MA Template	Victor (and Victor) defeated in ConflictOrElection at Place(and Place)
SF Prompt	⟨Vic_left0⟩Victor ⟨Vic_right0⟩(⟨Vic_left0⟩Victor ⟨Vic_right0⟩) ⟨Conf_left0⟩ConflictOrElection ⟨Conf_right0⟩ ⟨Place_left0⟩Place ⟨Place_right0⟩(⟨Place_left0⟩Place ⟨Place_right0⟩)
CA Template	Victor (Victor) ConflictOrElection Place (Place)

TABLE 4.1: Variants of prompt introduced in section 4.2.2. **MA**:Manual Template. **SF**:Soft Prompt. **CA**:Concatenation Template. Words with angle brackets in Soft Prompt denote role-specific pseudo tokens of continuous prompts. For multi-argument cases, we simply add slots within square brackets.

We first define text markers $\langle \mathbf{t} \rangle$ and $\langle /\mathbf{t} \rangle$ as special tokens then insert them into context X before and after the trigger word respectively.

$$\tilde{X} = [x_1, x_2, ..., \langle \mathbf{t} \rangle, x_{trig}, \langle /\mathbf{t} \rangle, ..., x_n]$$

Instead of concatenating the processed context $\tilde{X}$ and prompt Pt directly, we feed the context into BART-Encoder and the prompt into BART-Decoder separately, as illustrated in Figure 4.1. The prompt and context would interact with each other at the cross-attention layers in the decoder module.

$$\begin{aligned} H_X^{(enc)} &= \mathcal{L}_{enc}(\tilde{X}) \\ H_X &= \mathcal{L}_{dec}(H_X^{(enc)}; H_X^{(enc)}) \\ H_{pt} &= \mathcal{L}_{dec}(Pt; H_X^{(enc)}) \end{aligned} \tag{4.1}$$

where H_X denotes the event-oriented context representation and H_{pt} denotes context-oriented prompt representation. For k-th slot in the joint prompt we mean-pool its corresponding representations from h_{pt} and obtain role feature $\psi_k \in R^h$, where h denotes the dimension of hidden layer in BART. Note that a role may have multiple slots and, correspondingly, multiple role features and span selectors.

We adopt a simple but effective modification on previous QA-based methods by deriving **role-specific span selector** θ_k from every role feature in the prompt. Given role feature ψ_k, we have:

$$\begin{aligned} \psi_k^{(start)} &= \psi_k \circ w^{(start)} \in R^h \\ \psi_k^{(end)} &= \psi_k \circ w^{(end)} \in R^h \end{aligned} \tag{4.2}$$

where $\theta = [w^{(start)}; w^{(end)}] \in R^{h \times 2}$ is learnable parameters shared among all roles, and $\circ$ represents element-wise multiplication. $\theta_k = [\psi_k^{(start)}; \psi_k^{(end)}]$ is exactly the span selector for k-th slot in the prompt. With only one meta-head θ and simple operations, our method enables to generate arbitrary number of role-specific span selectors to extract related arguments from context. Recall the generation process of role feature ψ_k from prompt h_{pt}, it is obvious that both the interaction among different roles and the information aggregation between context and roles are considered under this paradigm.

4.2.4 Learning with Prompted Span Selector

Given context representation H_X and a set of span selectors $\{\theta_k\}$, each θ_k aims to extract at most one corresponding argument span (s_k, e_k) from H_X. For θ_k relating to one argument $a_k = \tilde{X}_{i:j}$, where i and j are the start and end word indices in context, the selector is expected to output $(\hat{s}_k, \hat{e}_k) = (i, j)$ as prediction. And for θ_k relating to no argument (when context has no argument about this role, or the slot number of this role exceeds the argument number), it is expected to output $(\hat{s}_k, \hat{e}_k) = (0, 0)$ representing an empty argument ϵ.

We first follow the formulation in Section 4.2.1 to calculate the distribution of each token being selected as the start/end of the argument for each role feature.

$$\begin{aligned} \text{logit}_k^{(start)} &= \psi_k^{(start)} H_X \in R^L \\ \text{logit}_k^{(end)} &= \psi_k^{(end)} H_X \in R^L \end{aligned} \tag{4.3}$$

where $\text{logit}_k^{(start)}$ and $\text{logit}_k^{(end)}$ represent start and end position distributions over the context tokens for each slot k, and L denotes the context length.

Then we calculate probabilities where the start/end positions locate:

$$\begin{aligned} p_k^{(start)} &= \text{Softmax}(\text{logit}_k^{(start)}) \in R^L \\ p_k^{(end)} &= \text{Softmax}(\text{logit}_k^{(end)}) \in R^L \end{aligned} \tag{4.4}$$

and define the loss function as:

$$\begin{aligned}\mathcal{L}_k(X) &= -(\log p_k^{(start)}(s_k) + \log p_k^{(end)}(e_k)) \\ \mathcal{L} &= \sum_{X \in D} \sum_k \mathcal{L}_k(X)\end{aligned} \tag{4.5}$$

where D ranges over all context in dataset and k ranges over all slots in prompt for X.

Bipartite Matching We optionally introduce bipartite matching to deal with multiple arguments of the same role for finding the global-optimal assignments with the least-cost match. Since we insert multiple slots about this role and each slot generates one prediction, it is a canonical bipartite matching problem that matches local-optimal predictions (of each slot) and ground truth as much as possible. Following previous work [208, 209], we use Hungarian algorithm [210]. Given $\text{logit}_k^{(start)}$ and $\text{logit}_k^{(end)}$, we apply greedy search on predicted start and end position distributions to select the predicted span for each role-specific selector θ_k.

$$(\hat{s}_k, \hat{e}_k) = \underset{(i,j) \in L^2, i<j}{\arg\max} \ \text{logit}_k^{(start)}(i) + \text{logit}_k^{(end)}(j) \tag{4.6}$$

Denote $y_r = \{(s_i, e_i)\}_{i=1}^n$ as ground truth spans of role r for sample X, and $\hat{y}_r = \{(\hat{s}_i, \hat{e}_i)\}_{i=1}^m$ as predicted spans, where m is the number of occurrence of role r in the corresponding prompt.

With the candidate spans for each role, we define the bipartite matching between the candidates and ground truth annotations as finding the lowest cost of a permutation Γ of N elements:

$$\hat{\sigma} = \underset{\sigma \in \Gamma_N}{\arg\min} \sum_k^N L_1((s_k, e_k), (\hat{s}_{\sigma(k)}, \hat{e}_{\sigma(k)})) \tag{4.7}$$

where $L_1((s_k, e_k), (\hat{s}_{\sigma(k)}, \hat{e}_{\sigma(k)}))$ represents L_1-norm between (s_k, e_k) and $(\hat{s}_{\sigma(k)}, \hat{e}_{\sigma(k)})$.

We introduce the Hungarian algorithm [210] for efficient optimal assignment. In Equation 4.7, N is chosen to the minimum value between m and n. If the number of candidate spans m is larger than the number of ground truth span n, we will pad $(0, 0)$ representing no arguments to the golden answer set. Otherwise, we only select the optimally matched gold spans for bipartite matching loss calculation.

After finding the optimal assignment $\hat{\sigma}$, we align each ground truth span in y_r and each predicted span in $\hat{y}_r$ according to the matching result and then calculate probabilities where the start/end positions locate about role slot k. Note that we use the logit distribution of $\hat{\sigma}(k)$ rather than k, which is different from Equation 4.4.

$$\begin{aligned} p_k^{(start)} &= \text{Softmax}(\text{logit}_{\hat{\sigma}(k)}^{(start)}) \\ p_k^{(end)} &= \text{Softmax}(\text{logit}_{\hat{\sigma}(k)}^{(end)}) \end{aligned} \tag{4.8}$$

Given $p_k^{(start)}$ and $p_k^{(end)}$ obtained by Equation 4.8, we follow the same loss function in Equation 4.5 during training process. The bipartite matching is only applied in training. For inference, the model will output all non-zero spans with corresponding argument roles as predictions.

4.2.5 Inference

For inference, we define the set of candidate spans for event arguments as $\mathcal{C} = \{(i,j)|(i,j) \in L^2, 0 < j - i \leq l\} \cup \{(0,0)\}$. It contains all spans shorter than the threshold l **and** special span $(0,0)$ indicating no arguments. Our model extracts the argument of each θ_k by enumerating and scoring all candidate spans as:

$$\text{score}_k(i,j) = \text{logit}_k^{(start)}(i) + \text{logit}_k^{(end)}(j) \tag{4.9}$$

and the predicted span of slot k is given by:

$$(\hat{s_k}, \hat{e_k}) = \underset{(i,j)\in\mathcal{C}}{\arg\max}\, \text{score}_k(i,j) \tag{4.10}$$

Since at most one span is predicted by each slot in the prompt, this strategy avoids the exhaustive threshold tuning.

4.3 Experiments

In this section, we explore the following questions: (1) Can PAIE better utilize PLMs for joint extraction to boost the performance of S-EAE and D-EAE? (2) How does PAIE address the challenges including few-shot and large-scale setting?

4.3.1 Experimental Setup

Datasets We conduct experiments on three common datasets in Event Argument Extraction task: ACE05 [196], RAMS [211] and WIKIEVENTS [89]. Regarding ACE05 [196], we follow the pre-processing procedure of DyGIE++ [212]. It keeps 33 event types and 22 argument roles and collects 4859 arguments in the training set, 605 and 576 in the development and test set respectively. RAMS [211] is a document-level dataset annotated with 139 event types and 65 semantic roles. Each sample is a 5-sentence document, with trigger word indicating pre-defined event type and its argument scattering among the whole document. WIKIEVENTS [89] is another document-level dataset providing 246 documents, with 50 event types and 59 argument roles. These documents are collected from English Wikipedia articles that describe real-world events and then follow the reference links to crawl related news articles. Table 4.2 shows their detailed statistics.

Dataset	ACE05	RAMS	WIKIEVENTS
#Sents			
Train	17,172	7,329	5,262
Dev	923	924	378
Test	832	871	492
#Args			
Train	4,859	17,026	4,552
Dev	605	2,188	428
Test	576	2,023	566
#Event	33	139	50
#Role	22	65	59
#Arg per Event	1.19	2.33	1.40

Table 4.2: Statistics of datasets.

Evaluation Metric We adopt two evaluation metrics. (1) Argument Identification F1 score (Arg-I): an event argument is correctly identified if its offsets and event type match those of any of the argument mentions. (2) Argument Classification F1 score (Arg-C): an event argument is correctly classified if its role type is also correct. For WIKIEVENTS dataset, we follow [89] and additionally evaluate Head F1 score (Head-C), concerning only the matching of the headword of an argument.

Implementation Details We initialize the weight in encoder-decoder architecture with pre-trained BART models. The contexts in the document-level dataset sometimes exceed the constraint of BART-Encoder and consume prohibitively large

Hyperparameter	**Value**
Batch size	16 (ACE05) / 4 (Others)
Weight decay	0.01
Training steps	10000
Optimizer	AdamW
Adam ϵ	1×10^{-8}
Adam β_1/β_2	0.9 / 0.999
Scheduler	Linear (with 0.1 warmup step)
Max span length	10
Max gradient norm	5.0
Window size	250
Max encoder seq length	192 (ACE05) / 500 (Others)
Max decoder seq length	80

TABLE 4.3: Hyperparameters for PAIE

memory; thus we add a window centering on the trigger words and only encode the words within the window. We train each large model on single NVIDIA-V100 GPU and each base model on a single NVIDIA-1080Ti GPU. For each setting, we train models with 5 fixed seeds [13, 21, 42, 88, 100] and 3 learning rates [2e-5, 3e-5, 5e-5]. Then we record the test set performance of the model that performs best on the development set for each random seed. The final reported performance is the average value of results w.r.t five different seeds. We list other important hyperparameters in Table 4.3.

Baselines We compare our model with following previous models. (1) **ONEIE** [213]: a joint model extracting entity, relation and event simultaneously. Different from QA-based model, they rely on extracted entities as candidate arguments. (2) **BART-Gen** [89]: a conditional generation model generating (rather than recognizing the spans) arguments sequentially via a sequence-to-sequence model and prompt. (3) **EEQA** [68]: the first Question Answering (QA) based model designed for sentence-level EAE task. (4) **FEAE** [87]: a QA-based method extended to document-level EAE by considering argument interactions via knowledge distillation. (5) **DocMRC** [88]: another QA-based method with implicit knowledge transfer and explicit data augmentation.

For the models we re-trained, we keep all other hyper-parameters except learning rates the same with default settings in their original papers. We search the learning rate in [2e-5, 3e-5, 5e-5] and report the test set performance of the model that performs best on the development set.

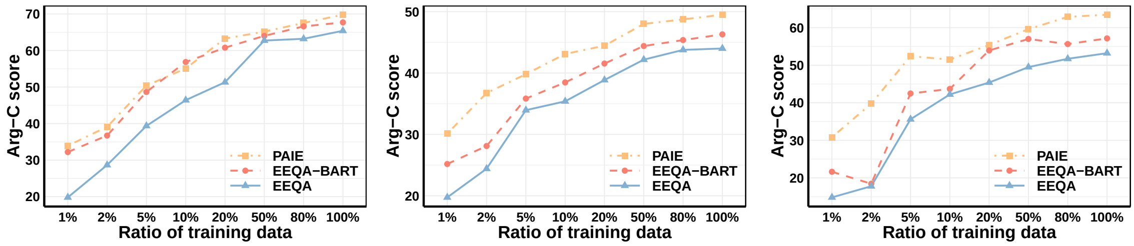
Model	PLM	ACE05		RAMS		WIKIEVENTS		
		Arg-I	**Arg-C**	**Arg-I**	**Arg-C**	**Arg-I**	**Arg-C**	**Head-C**
FEAE [87]	BERT-b	-	-	53.5*	47.4*	-	-	-
DocMRC [88]	BERT-b	-	-	-	45.7*	-	43.3*	-
OneIE [213]	BERT-b	65.9	59.2	-	-	-	-	-
	BERT-l	73.2	69.3	-	-	-	-	-
EEQA [68]	BERT-b	68.2*	65.4*	46.4	44.0	54.3	53.2	56.9
	BERT-l	70.5	68.9	48.7	46.7	56.9	54.5	59.3
BART-Gen [89]	BART-b	59.6	55.0	50.9	44.9	47.5	41.7	44.2
	BART-l	69.9*	66.7*	51.2	47.1	66.8	62.4	65.4
EEQA-BART	BART-b	69.6	67.7	49.4	46.3	60.3	57.1	61.4
	BART-l	73.1	<u>72.2</u>	51.7	48.7	61.6	57.4	61.3
PAIE (Ours)	BART-b	<u>73.6</u>	69.8	<u>54.7</u>	<u>49.5</u>	<u>68.9</u>	<u>63.4</u>	<u>66.5</u>
	BART-l	**75.7**	**72.7**	**56.8**	**52.2**	**70.5**	**65.3**	**68.4**

TABLE 4.4: Overall performance. We highlight the best result and underline the second best. * means the value from the original paper. **b** in column **PLM** denotes base model and **l** denotes large model.

4.3.2 Overall Performance

Table 4.4 compares our approach with all baselines. We observe that PAIE performs best on all datasets. For S-EAE, our base model achieves an absolute Arg-C improvement of 2.1% on ACE05. For D-EAE, our base model obtains 2.1% and 6.3% Arg-C gains on RAMS and WIKIEVENTS, respectively. Similarly, our large-version model achieves 3.5% and 2.9% gains. This demonstrates a good generalization ability of our proposed method on dealing with varying lengths of context.

We also find that QA-based model sometimes performs well even in document-level EAE tasks. The EEQA-BART model shows almost the same Arg-C with BART-Gen [89] on RAMS dataset. Other QA-based models (especially those considering argument interactions, like FEAE [87]) also have competitive performance. As for WIKIEVENTS, however, QA-based models are inferior to sequential-generation models significantly. We speculate that the performance of previous QA-based models is not robust to handle longer text. Both BART-Gen [89] and our model PAIE have a relatively stable performance on various document-level EAE datasets, but our model performs better, especially with smaller PLMs. Next, we conduct further analysis with the strongest baseline EEQA-BART and our PAIE. We use the base-version BART for a fair comparison.



(A) ACE05 (B) RAMS (C) WIKIEVENTS

FIGURE 4.2: Arg-C F1 scores w.r.t different training data ratios.

Model	**ACE05** B	**ACE05** L	**RAMS** B	**RAMS** L	**WIKI** B	**WIKI** L
BART-Gen	5.8	12.4	33.2	54.8	19.1	29.0
EEQA-BART	11.8	36.0	66.0	187.4	30.9	83.8
PAIE	**2.9**	**8.4**	**19.0**	**38.6**	**8.4**	**18.3**

TABLE 4.5: Inference time (second) for different models on test set of three benchmarks. Experiments are run on single NVIDIA-1080Ti GPU.

4.3.3 Few-shot Setting

We analyze how PAIE performs under a scenario without sufficient annotations. Figure 4.2 shows the performance of PAIE and two other QA-based baselines with partial training samples on three benchmarks. It demonstrates that **(1)** PAIE is superior to EEQA-BART and EEQA in almost all settings with different datasets and training data ratios. **(2)** PAIE especially outperforms QA-based methods in document-level tasks (RAMS and WIKIEVENTS). It achieves comparable F1 scores with EEQA-BART using only about 20% training samples and EEQA using about 10% samples. **(3)** Along with the decreasing number of training data, the gains become larger than baselines. All observations above indicate that PAIE can better utilize PLMs for few-shot settings.

4.3.4 Large-scale Setting

We analyze how PAIE performs when there are large amounts of documents to be extracted. Results in Table 4.5 demonstrates that PAIE also has much better extraction efficiency compared with other approaches. We report the overall inference time for different models. PAIE usually runs 3-4 times faster than EEQA, since it predicts multiple roles simultaneously, while EEQA predicts roles one by one. Other QA-based models are likely to have similar speeds with EEQA due to their sequential prediction structure and training process. Also, as discussed in Section 4.4.5, PAIE is even more advantageous under practical application scenarios since it avoids the heavy threshold tuning.

4.4 Analysis

4.4.1 Ablation Study

In this section, we investigate the effectiveness of our main components by removing each module in turn. (1) **bipartite matching**. We drop out of the bipartite matching loss and ignore the global optimal span assignment. (2) **multi-arg prompt**. We additionally replace the prompt containing multiple roles with several single templates in which include only one role. (3) **role-specific selector**. The selector is not role-specific anymore but is shared among all roles. This variant degrades to EEQA-BART.

We summarize the results of ablation studies in Table 4.6. **(1)** EEQA-BART outperforms EEQA significantly, which demonstrates that even conventional QA-based methods have substantial space for improvement with a better PLM and span selection strategy. **(2)** The role-specific selector further improves Arg-C scores in RAMS and WIKIEVENTS, while taking a slightly negative effect on ACE05. Since the former two datasets are document-level and have more role types (65 in RAMS, 59 in WIKIEVENTS, and 36 in ACE05), we speculate that role-specific selector plays a critical role when identifying and disambiguating roles with complicated ontology structures in long documents. **(3)** Joint multi-argument prompt achieves consistent improvement on all three datasets. It indicates that the joint prompt has the potential to capture implicit interaction among arguments. **(4)** Bipartite matching loss has an average improvement of 0.7% on three benchmarks. We conjectured it is due to the permutation-invariance property of bipartite matching.

Model	Bipartite Matching	Multi-arg Prompt	Role-specific Selector	PLM	Arg-C ACE05	Arg-C RAMS	Arg-C WIKI
PAIE	✓	✓	✓	BART-b	69.8$_{\pm 0.98}$	49.5$_{\pm 0.65}$	63.4$_{\pm 1.17}$
PAIE$_{\text{-w/o bipartite}}$	✗	✓	✓	BART-b	68.9$_{\pm 1.03}$	49.4$_{\pm 0.98}$	62.4$_{\pm 1.09}$
PAIE$_{\text{-w/o multi-prompt}}$	✗	✗	✓	BART-b	66.9$_{\pm 0.61}$	47.6$_{\pm 1.20}$	59.9$_{\pm 1.26}$
EEQA-BART	✗	✗	✗	BART-b	67.7$_{\pm 0.64}$	46.3$_{\pm 0.77}$	57.1$_{\pm 0.82}$
EEQA	✗	✗	✗	BERT-b	65.4	44.0	53.2

TABLE 4.6: Ablation study on three benchmarks.

4.4.2 Architecture Variants

PAIE feeds the context into BART-Encoder and the prompt into BART-Decoder respectively. A plausible and straightforward variant called **PAIEE** (PAIE-Encoder) concatenates context and prompt, then feed them into encoder directly. We investigate the performance of PAIEE compared with PAIE in this section, as shown in Table 4.7.

We can see that concatenating context and prompt slightly impairs the model performance. It seemingly indicates that the over-interaction between context and prompt is not of benefit. Furthermore, the prompt squeezes the limited input length of the encoder kept for a document if it concatenates with the document. The experiments support our strategy feeding context and prompt separately without concatenation to PAIE.

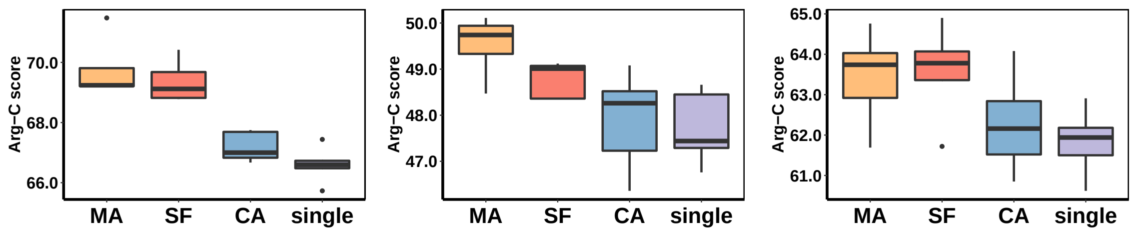
Variant	PLM	ACE05	RAMS	WIKI
PAIEE	BE-b	65.9	46.3	62.9
	BA-b	70.2	49.3	62.8
	BA-l	<u>72.3</u>	<u>51.7</u>	<u>65.1</u>
PAIE	BA-b	69.8	49.5	63.4
	BA-l	**72.7**	**52.2**	**65.3**

TABLE 4.7: Arg-C F1 of different PLMs. BE and BA denote BERT and BART. Note that we also try PLM with only encoder such as BERT under PAIEE setting, which does not require a decoder.

4.4.3 Prompt Variants

We investigate how different types of prompts affect the performance in this section, as shown in Figure 4.3. We compare four different prompts: three joint prompts introduced in Section 4.2.2 and one single template containing only one role slot, i.e. the question template used in QA-based method.

We find that **(1)** All three joint prompts outperform the single template, which validates the effectiveness of the joint prompt. **(2)** Manual template has the most stable performance and usually the better result than others. **(3)** Soft prompt achieves comparable result with a manual template. We claim this observation inspiring because the creation of the manual template is laborious and soft prompts almost avoid such a handcrafted process. It also accords with current trends of creating distinct continuous prompts, which usually perform better than manual ones. **(4)** Concatenation template performs worst among joint prompts. We conjecture it is due to such prompt neither contains prior knowledge about role interaction (manual template) nor learns such interaction during training (soft prompt).



(A) ACE05 (B) RAMS (C) WIKIEVENTS

FIGURE 4.3: Arg-C F1 using three different types of joint prompts in Table 4.1 plus the single template on three benchmarks. **MA**: Manual Template. **SF**: Soft Prompt. **CA**: Concatenate Template. **single**: Single Template.

4.4.4 Long-range Dependencies

Model	Trigger-Argument Distance d				
	$-2_{[79]}$	$-1_{[164]}$	$0_{[1811]}$	$1_{[87]}$	$2_{[47]}$
BART-Gen	17.7	16.8	44.8	16.6	9.0
DocMRC	21.0	20.3	46.6	17.2	12.2
FEAE	**23.7**	19.3	49.2	25.0	5.4
EEQA-BART	15.6	<u>24.0</u>	51.7	23.5	8.0
PAIE$_{\text{w/o multi-prompt}}$	21.2	21.4	<u>52.3</u>	<u>27.9</u>	<u>24.6</u>
PAIE	<u>21.7</u>	**27.3**	**54.7**	**29.4**	**25.4**

TABLE 4.8: Performance (Arg-C F1 score) breakdown by argument-trigger distance d on RAMS development set. The argument number of each case is given in the bracket.

Model	WIKIEVENTS Argument Number n			
	$\mathbf{1}_{[468]}$	$\mathbf{2}_{[66]}$	$\mathbf{3}_{[15]}$	$\geq \mathbf{4}_{[17]}$
EEQA-BART	$58.0_{(-16)}$	$59.7_{(-3)}$	$28.6_{(-10)}$	$10.0_{(-26)}$
PAIE	**74.1**	**62.6**	**38.1**	**36.4**

TABLE 4.9: Arg-C F1 on WIKIEVENTS breakdown by argument number n of one role. The case number is given in the square bracket.

In D-EAE task, arguments could span multiple sentences. Therefore, the model is required to capture long-range dependencies. For better evaluating PAIE and comparing with others, we list their performance breakdown on different sentence distances between arguments and the given trigger word in Table 4.8. We can see that **(1)** PAIE significantly improves the ability to extract arguments with long distances, especially for those behind the trigger words (see columns with positive d values). **(2)** The last two rows of the table indicate that joint prompts in PAIE leverage the implicit interaction among roles, and roles conditioning on each other lower the difficulty to extract long-distance arguments effectively.

4.4.5 Same-role Argument Assignment

Multiple arguments may share the same role in the same event. We show that PAIE outperforms QA-based models dealing with it in both few-shot and large-scale settings.

Few-shot We also compare the capability of PAIE and EEQA-BART in predicting multiple arguments with the same role on WIKIEVENTS, a dataset containing diverse multi-argument cases. Table 4.9 shows that PAIE outperforms significantly better than EEQA-BART dealing with such cases. For roles with three and four or more arguments, PAIE gains a definite Arg-C F1 improvement of 9.5% and 26.4%, respectively.

Large-scale To solve this problem, QA-based methods usually adopt the thresholding strategy, which compares the score of each text span with a manually tuned threshold. We claim that it consumes lots of time and computational resources for finding a good threshold and usually ends with sub-optimal results. We support such claim by a coarse grid search tuning span threshold on WIKIEVENTS dataset using EEQA and EEQA-BART models, as shown in Figure 4.4. The choice

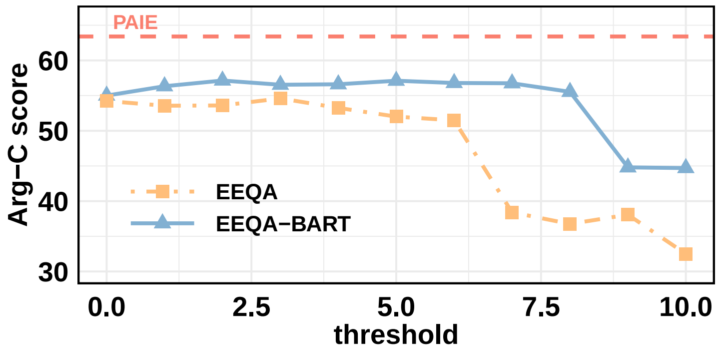


FIGURE 4.4: Arg-C F1 w.r.t different thresholds for WIKIEVENTS. We draw the performance of PAIE in red dashed line for comparison (no threshold tuning).

of threshold highly affects the performance of the model. In addition, models with the same architecture but different PLMs have totally different optimal thresholds even on the same dataset, not to mention on distinct datasets. PAIE requires no threshold tuning since each slot in the prompt only predicts at most one argument span and usually achieves much higher inference speed in practice.

4.5 Conclusion

Towards the better few-shot and large-scale EAE algorithm, we propose a novel approach PAIE that effectively and efficiently extracts arguments at both sentence and document levels. We define a new prompt tuning paradigm for extraction tasks, which prompts multiple role knowledge from PLMs via role-specific selectors and joint prompts. Extensive experiments on three standard benchmarks demonstrate our proposed model's promising capabilities on both few-shot and large-scale settings of EAE task. We have also conducted ablation studies on the main components, the extractive prompting strategy, and several real scenarios. In the future, we are interested in investigating co-reference as an auxiliary task of EAE and introducing entity information to better determine argument boundaries.

Chapter 5

Filter-then-rerank Paradigm for Information Extraction

In previous chapters, we explore the use of small language models (SLMs) for information extraction (IE). This chapter [1] shifts focus to large language models (LLMs, [44, 46]) and pursue their applications in IE tasks under **few-shot** and **large-scale** settings. Though our pivot experiments show that LLMs generally underperform SLMs in IE efficiency and effectiveness, deeper analysis reveals that LLMs excel at handling challenging samples. Based on this finding, we propose a *filter-then-rerank* paradigm where SLMs (filter) are used for initial predictions and LLMs (reranker) are selectively applied to difficult cases. Extensive experiments validate that it significantly beats all competitive baselines under **few-shot** settings and reduces LLM-calling cost under **large-scale** scenarios.

5.1 Introduction

Large Language Models (LLMs, [44, 46]) have shown remarkable abilities on various NLP applications such as factual question answering [214, 215], arithmetic reasoning [216, 217] and logical reasoning [218, 219]. Given the reasoning, memorization, instruction-following and few-shot adaption capabilities emerging from

[1] This chapter is published as Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. "Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!". Findings of the Association for Computational Linguistics: EMNLP 2023, pages 10572–10601, Singapore, 2023. [36]

LLMs, it prompts a compelling question: Can LLMs be used to boost performance of IE tasks under few-shot and large scale settings (in other words, handling IE tasks in an effective and efficient approach)?

To answer this question, we conduct an extensive empirical study to compare the performance between LLMs using *in-context learning* [2] (ICL [220]) and *fine-tuned* Small Language Models (SLMs). We fairly evaluate SLMs-based and LLMs-based methods across nine datasets spanning four common IE tasks: (1) Named Entity Recognition, (2) Relation Extraction, (3) Event Detection and (4) Event Argument Extraction. For each dataset, we explored four to six settings to encompass typical low-resource extents, from 1-shot to 20-shot or even more. Given the potential sensitivity of LLMs' performance to the prompt context, we meticulously considered variations in instruction, demonstration number and selection strategy, prompt format, *etc.*. Our study reveals that LLMs excel over SLMs only when annotations are extremely limited, *i.e.,* both label types[3] and the samples[4] per label are extremely scarce. With more (*e.g.,* hundreds of) samples, SLMs significantly outperform LLMs. Furthermore, LLMs incur greater inference latency and costs than fine-tuned SLMs. Hence, we claim that **current LLMs are not good few-shot and large-scale information extractors in general**.

Given the great potentials of LLMs, we further investigate whether LLMs and SLMs exhibit different abilities to handle various types of samples. We categorize samples according to their difficulty measured by SLMs' confidence scores, and compare LLMs' and SLMs' results within each group. We find that **LLMs are good at hard samples, though bad at easy samples**. We posit that the knowledge and reasoning abilities in LLMs enable them to handle hard samples (which are simply beyond SLMs' capabilities) well. Nevertheless, LLMs demonstrate strong predisposition to false-positive predictions on negative samples. Since most negative samples are easy samples (which could be solved readily by SLMs), the performance of LLMs on easy samples sometimes collapses and are usually much worse than fine-tuned SLMs.

[2]All LLMs discussed in this chapter are not fine-tuned, and results for LLMs are based on in-context learning.

[3]Label types denote *entity/relation/event/role types* in different tasks. We use them interchangeably there-in-after.

[4]Samples refer to (i) demonstrations in ICL of LLMs, or (ii) training samples for SLMs' fine-tuning.

Leveraging these findings, we pursue an approach to incorporate LLMs and SLMs within a single system and combine their merits. To this end, we propose a novel *filter-then-rerank* framework. The basic idea is that SLMs serve as a filter and LLMs as a reranker. Specifically, SLMs initially predict and determine the difficulty of each sample. If the sample is a hard one, we further pass the top-N most-likely candidate labels from SLMs to LLMs for reranking. Otherwise we view the prediction from SLMs as the final decision. By providing easy/hard samples with different solution strategies, our system utilizes each model's strengths to complement each other. Also, it reranks only a small subset of samples and minimizes the extra latency and budgets for calling LLMs. This design makes our *filter-then-rerank* friendly under large-scale scenarios. With a modest cost increase, our framework yields a consistent F1 improvement, averaging 2.4% higher than previous methods on various few-shot IE tasks. To the best of our knowledge, this is the first successful attempt to use LLMs to enhance few-shot IE tasks.

5.2 Large LMs v.s. Small LMs

In this section, we compare the performance between LLMs and SLMs to evaluate whether LLMs perform competitively. Our empirical study reveals that LLMs are not generally good information extractors under few-shot and large-scale settings.

5.2.1 Task, Dataset and Evaluation

We run experiments on nine widely-used datasets across four IE tasks. (1) Named Entity Recognition (NER): CONLL03 [221], OntoNotes [222] and FewNERD [223]. (2) Relation Extraction (RE): TACRED [224] and TACREV [225]. (3) Event Detection (ED): ACE05 [196], MAVEN [197] and ERE [198]. (4) Event Argument Extraction (EAE): ACE05, ERE and RAMS [211]. With label numbers ranging from 4 to 168, we assess LLMs' performance under different schema complexities.

Few-shot Set We construct few-shot datasets from the original datasets above. For training and validation set, we adopt K-shot sampling strategy, *i.e.,* sampling K samples for each label type. For test set, we downsample their original test sets to reduce the cost of LLMs. We randomly sample 500 sentences for RE tasks, and 250 sentences for other task. We ensure that each label has at least one corresponding sample to avoid the absence of rare labels.

Evaluation We adopt micro-F1 score in NER, RE and ED tasks. For EAE task, we follow previous work [226] and adopt head-F1 score, which merely considers matching of the head word rather than the whole content of a text span. We report averaged score w.r.t 5 sampled train/validation sets unless otherwise stated.

5.2.2 Small Language Models

We adopt five supervised methods to evaluate the abilities of SLMs. (1) Vanilla fine-tuning for all tasks, (2) FSLS [58] for NER and ED tasks, (3) KnowPrompt [64] for RE task, (4) PAIE [33] for EAE task, and (5) UIE [74] for all tasks.

5.2.3 Large Language Models

We evaluate the ICL abilities of LLMs in this section. Given labeled sentences $D = \{(s_i, y_i)\}$ and a test sentence s, our goal is to predict structured information y from s using a frozen LLM $\mathcal{L}$. We feed LLM with prompt $\mathcal{P}_{\mathcal{E},I,f}(D, s)$:

$$\mathcal{P}_{\mathcal{E},I,f}(D, s) = [I; f(\mathcal{E}(D, s)); f(s)] \tag{5.1}$$

We give examples of prompts on four IE tasks in Figure 5.1. The prompts consist of three parts: instruction I (color in green in Figure 5.1), demonstration $f(\mathcal{E}(D, s))$ (demo; color in blue) and the question $f(x)$ (color in black). Here $\mathcal{E}$ denotes demo selector and $\mathcal{E}(D, s) \subset D$ denotes selected sentences as the demo to predict s. Prompt format f [5] refers to the template which converts demo $\mathcal{E}(D, s)$ and sample s to input context for LLMs. Then LLM generates $f(y)$ (color in red) from which we could readily parse the extraction results y.

[5] We slightly abuse the notation f to allow s, y and $\{(s, y)\}$ as the input for simplicity.

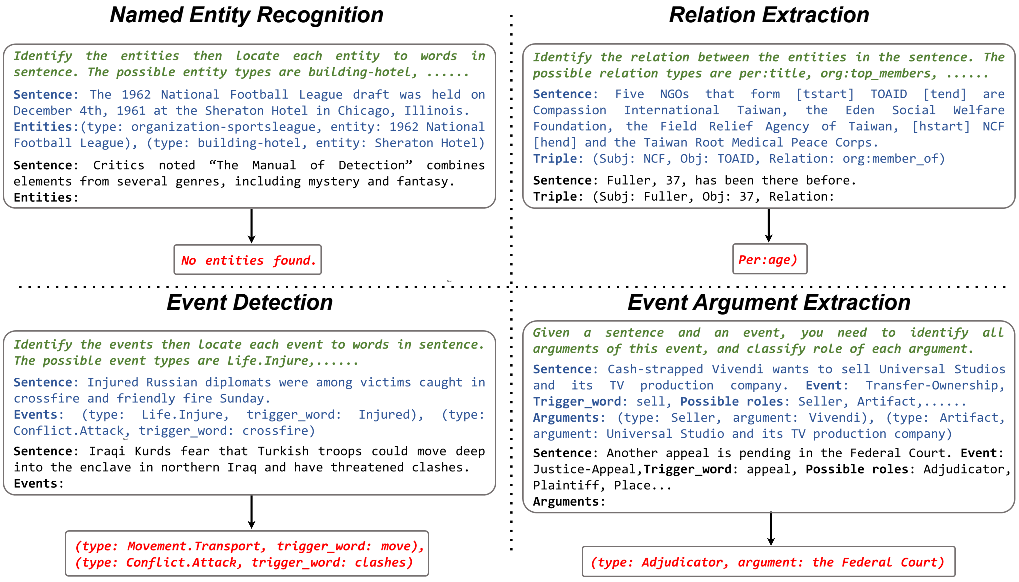


FIGURE 5.1: Examples of prompts used. The green, blue and black parts in the top boxes represent the instruction, demonstration (demo) and test sentence in the prompt respectively. The red parts represent the outputs from LLMs. We plot only 1 example for convenience of visualization. The actual demo number is usually much larger than 1.

Models $\mathcal{L}$: We explore six LLMs from two sources. (1) OpenAI models [6]: we employ ChatGPT, CODEX [45] and InstructGPT [227] for main experiments (2) Open-source models: we use LLaMA-13B [46] and its instruction-tuned counterpart, Vicuna-13B [228].

Instruction I: The instruction (1) describes the task and (2) enumerates all possible labels for reference. we adopt instructions shown in Figure 5.1.

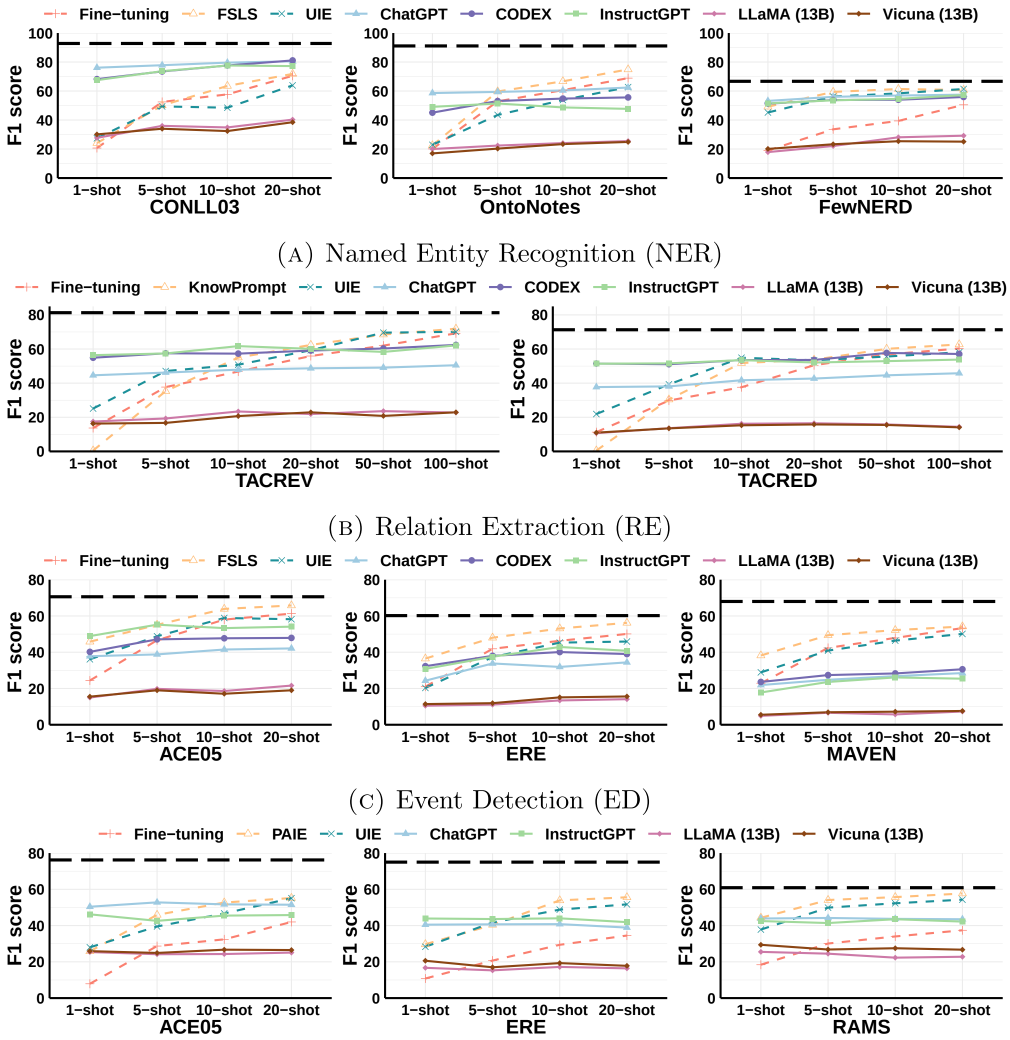
Demo selector $\mathcal{E}$: The maximum input length of LLMs usually limits the sentence number in demos even under few-shot settings. Therefore for each test sentence s, we demand a demo retriever $\mathcal{E}(D, s)$ which selects a small subset from D as the sentences in demo. Following previous methods [229, 230], we retrieve demos according to their sentence embedding similarity to the test samples.

Prompt format f: We use simple textual templates to format the demos and the test sample in main experiments. For example, the template for NER is

[6]The versions of model we use are: `gpt-3.5-turbo-0301`, `code-davinci-002`, `text-davinci-003` and `gpt-4-0314`. Due to budget constraints, we execute InstructGPT and GPT-4 only once per setting. We do not conduct EAE task on CODEX since it had been unavailable at that time.

```
Sentence:  [S]
Entities:  ([type1], [entity1]), ([type2], [entity2])...
```

5.2.4 Main Results



(A) Named Entity Recognition (NER)

(B) Relation Extraction (RE)

(C) Event Detection (ED)

(D) Event Argument Extraction (EAE)

FIGURE 5.2: Overall results of SLM-based methods (dashed lines) and LLM-based methods (solid lines) on nine datasets across four IE tasks. The black, horizontal dashed lines represent the SoTA performance on full dataset.

We summarize the main experimental outcomes in Figure 5.2, indicating that LLMs only outperform SLMs in environments with restricted labels and samples. Conversely, SLMs are generally more effective. Given (1) the practicality of fine-grained IE tasks and the manageable effort of obtaining 10-20 annotations per label and (2) the excessive time and budget demands of LLM inference, we conclude that LLMs are not as effective as supervised SLMs for few-shot IE tasks under real scenarios. We detail our findings as below.

Performance w.r.t sample number. The performance dynamics of SLMs and LLMs are influenced by variations in sample size. Under extremely low-resource (1-shot or 5-shot) settings, LLMs sometimes present superior performance than SLMs. Yet, LLMs tend to reach a performance plateau with only modest increases in sample size. Conversely, SLMs demonstrate marked performance enhancement as sample sizes grow. This trend is evident in Figure 5.2, where SLM trajectories (dashed lines) ascend more steeply compared to LLM ones (solid lines).

Performance w.r.t label number. Compared with SLMs, LLMs tend to struggle on fine-grained datasets. For instance, LLMs perform *relatively* worse on MAVEN and RAMS datasets (with 168/139 labels) than on CONLL (4 labels only). It illustrates a clear negative correlation between the label number and the result disparity between LLMs and SLMs across various IE tasks.

Comparisons among LLMs. We observe performance variability among LLMs. (1) Open-source models, LLaMA and Vicuna, significantly lag behind proprietary LLMs across all few-shot IE tasks. (2) Among proprietary LLMs, ChatGPT performs better on NER and EAE tasks, but poorer so on RE and ED tasks. InstructGPT and CODEX demonstrate comparable performance across these tasks.

Table 5.1: The inference seconds over 500 sentences (run on single V100 GPU). Here LLaMA is extremely slow since we set batch size as 1 due to memory limit.

Dataset (Task)	Roberta	T5	LLaMA	CODEX
FewNERD (NER)	2.8	39.4	1135.4	179.4
TACREV (RE)	1.4	45.6	1144.9	151.6
ACE05 (ED)	6.6	62.5	733.4	171.7

LLMs show limited inference speed. We compare the inference speed of different methods and show their results in Table 5.1. We observe that LLMs are much slower than SLMs since they have much more parameters, longer input contexts and extra response decay (if external APIs are applied). It shows that LLMs are not more suitable choices than SLMs for large-scale IE applications.

5.2.5 Analysis on Prompt Sensitivity

Previous work [231] indicates that the efficacy of LLMs on specific tasks can be significantly influenced by the construction of the prompt. To ensure that LLMs' suboptimal outcomes are not erroneously ascribed to inappropriate prompt designs, we meticulously examine the impact of diverse prompt variations from four aspects, *i.e.,* instruction format, demo number, demo selector and prompt format. We illustrate salient findings in Figure 5.3. Our findings include that (1) diverse instruction strategies yield comparable results in IE task; (2) increasing the number of samples in demonstrations does not unequivocally enhance performance; and (3) The selection strategy of demonstration matters, and retrieval based on sentence embedding (what we used) proves sufficiently effective. Consequently, we believe that there unlikely exists a *lottery* prompt that substantially alters our conclusions that LLMs are not good few-shot IE solver.

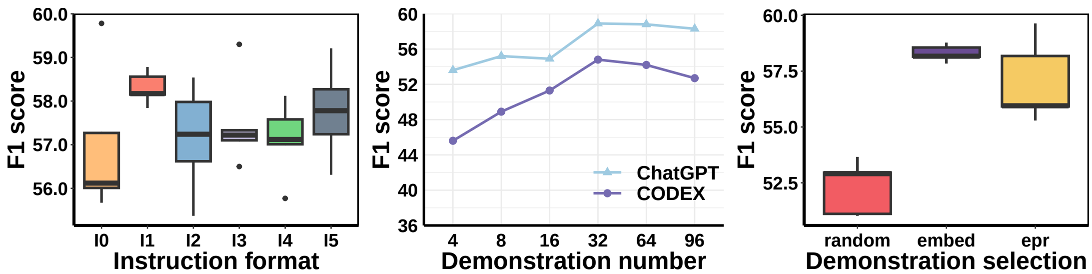


FIGURE 5.3: LLMs' performance w.r.t prompt variants on 20-shot FewNERD dataset. **Left**: ChatGPT's performance (F1 Score) across six instruction variants. **Middle**: F1 Score changes over varying numbers of demo. **Right**: ChatGPT's performance across three demo selection strategies. Random: Random sampling. Embed: Sentence embedding. EPR: Efficient Prompt Retriever [1].

5.2.6 Discussion: Why LLMs Fail to Obtain Satisfactory Performance on IE Tasks?

Underutilized Annotations. We notice that LLMs appear to benefit less from additional annotations, *i.e.,* more training samples and label types, than SLMs. We speculate that LLMs are constrained by ICL in two ways. (1) More samples: The number of effective samples for LLMs, those in demos, is limited by maximum input length. Moreover, we also observe LLMs' performance plateaus in some tasks before reaching this limit. Meanwhile, SLMs can continually learn from more samples through supervised learning, widening the performance gap as annotated samples increase. (2) More labels: LLMs struggle with fine-grained datasets. It suggests a difficulty in understanding numerous labels and their subtle interactions merely from the given instruction and exemplars for LLMs. Also, the examples per label in demos decrease as label types increase.

Unexplored Task format. As stated in Zhang et al. [232], IE-related tasks are scarce in the widely-used instruction tuning datasets like Chung et al. [122] and Wang et al. [233]. Furthermore, the highly-flexible format of NER and ED tasks impair the ICL abilities [7]. Therefore it is likely that instruction-tuned LLMs are not well-acquainted with such IE-related task formats.

5.3 LLMs are Good Few-shot Reranker

5.3.1 Filter-then-rerank Paradigm

To mitigate LLMs' drawbacks mentioned above, we propose a *filter-then-rerank* paradigm to integrate both SLMs and LLMs within the same system. This paradigm uses SLMs as filters to select the top-N candidate labels, then LLMs rerank them to make final decisions. By using SLM-generated candidate answers, the focus of LLMs shifts from **sentence-level** (*i.e.,* identifying all entities/events in the sentence) to **sample-level** (*i.e.,* determining single entity/event candidate provided). Each question now corresponds to a single sample, allowing us to reframe prompts as multi-choice questions (MCQ; shown in Figure 5.4) problem. Under such format, each candidate label is converted to a choice by pre-defined templates. We claim *filter-then-rerank* paradigm is more likely to elicit the powers of LLMs and smoothly solve few-shot IE tasks because: (1) LLMs are more familiar with MCQ prompts than IE-format prompts [232]. (2) This paradigm reduces the label scopes significantly, since N is usually much smaller than fine-grained label numbers.

[7] These two tasks require unfixed numbers of (label, span) tuple. Furthermore, the length of each span is also unfixed.

```
Read following sentences and identify what is the entity type
of "The New Yorker" quoted by <t>.
Sentence:
In 2004 Gourevitch was assigned to cover the 2004 U.S.
presidential election for "<t> The New Yorker <t>".
Candidate Choices:
(a)The New Yorker does not belong to any known entities.
(b)The New Yorker is a broadcast program.
(c)The New Yorker is a kind of written art.
(d)The New Yorker is a media/newspaper organization.
Analysis:
The New Yorker is a well-known American magazine that has
been published since 1925, and is primarily known for its
long-form journalism, commentary, and satire. It has a
reputation for publishing high-quality writing on a wide
variety of topics, including politics, culture, and the arts.
So The New Yorker is a media/newspaper organization.
Correct Answer: (d)
```

FIGURE 5.4: Multi-choice question (MCQ) prompt.

5.3.2 LLMs are *Hard* Sample Solver

Our *filter-then-rerank* paradigm, unfortunately, presents unsatisfactory performance (and even suffers longer latency since LLMs rerank candidates per sample). Given LLMs' abilities in memorization and reasoning, however, we still believe that LLMs are potential to solve **some**, if not most, IE samples effectively. We hypothesize that LLMs are more proficient than SLMs on *hard* samples. These samples are characterized by their requisite for external knowledge acquisition or sophisticated reasoning strategies, areas where LLMs can leverage their extensive parametric knowledge bases and inherent reasoning mechanisms. In contrast, SLMs often falter with such samples, constrained by their restricted modeling capacities.

We leverage an unsupervised metric from SLMs to evaluate the *difficulty* of samples. Given a sample x in the sentence s, we define the highest probability across all labels as the confidence score:

$$\text{conf}(x) = \max_{l \in L} P_{SLM}(l|x; s) \tag{5.2}$$

where L denotes the label set and $P_{SLM}(l|x; s)$ the probability of a span x (in the sentence s) referring to label l computed by SLMs. We classify samples with low confidence scores as *hard* samples. Otherwise we view them as easy samples.

We conduct experiments to confirm our hypothesis that LLMs excel on *hard* samples. We group samples by confidence scores and compare two methods within each group: (a) SLM-based methods without LLM reranking, and (b) SLMs as the filter and LLMs as the reranker. Method (b) differs from (a) by adding a single LLM to rerank the top-N SLM predictions, using MCQ prompts.

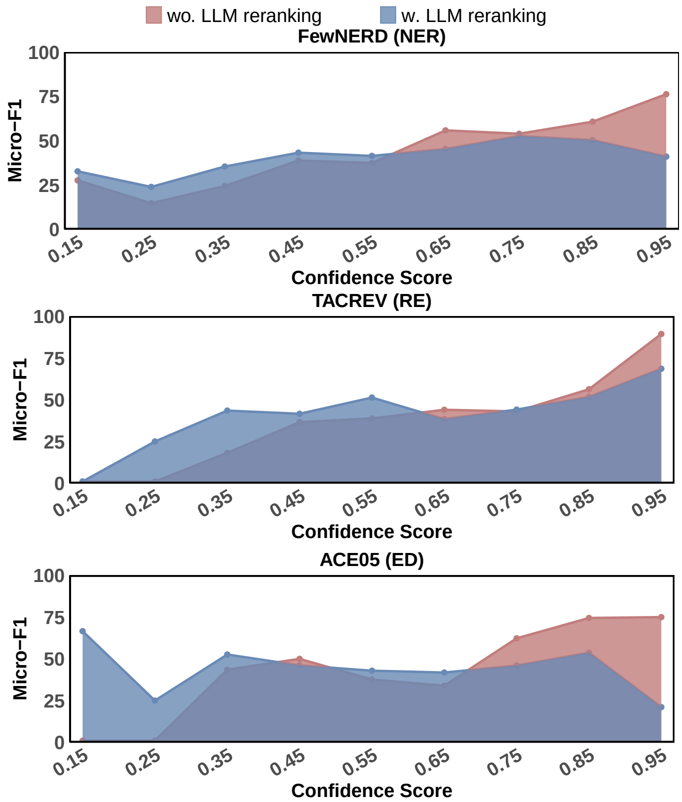


Figure 5.5: Relationship between confidence scores and performance with-/without LLM reranking. We adopt `RoBERTa-large` as filter and InstructGPT as reranker.

The results in Figure 5.5 support our assumption. (1) LLM-based reranking (blue lines) enhances performance on hard samples (left areas in the figure). We speculate that the efficacy of LLMs in harnessing external knowledge and complex reasoning to rectify erroneous predictions initially made by SLMs (red lines). (2) Conversely, LLM-based reranking impedes performance on easy samples (right areas), resulting in a significant degradation, particularly for very easy samples (rightmost areas). In conclusion, LLMs exhibit greater proficiency in handling hard samples compared to SLMs, yet they underperform relative to SLMs on easy samples.

Table 5.2: Comparative ratios of negative to positive samples across various datasets and subsets. We set fixed threshold τ here for simplicity.

	FewNERD	**TACREV**	**ACE05**
Overall	5.88	3.03	38.2
Easy samples ($\tau > 0.9$)	9.44	3.21	44.0
Hard samples ($\tau < 0.6$)	1.28	2.68	1.36

5.3.3 Why LLMs Fail on Easy Samples

We investigate why LLMs (relatively) fail on easy samples in this section. As shown in Table 5.2, we observe significant higher negative sample ratios for easy samples across diverse IE tasks. In other words, most negative samples are easy samples for SLMs. Here we refer negative samples to those labeled as `None`. We speculate that the proficiency of SLMs with negative samples stems from their ability to adeptly discern apparent patterns during the fine-tuning stages. Therefore, SLMs could predict negative samples with (relatively) high confidence and accuracy. Due to LLMs' predisposition to false-positive predictions on negative samples, however, the performance of LLMs on easy samples collapses. We attribute such false-positive predictions to (1) hallucination and (2) span boundary mismatch.

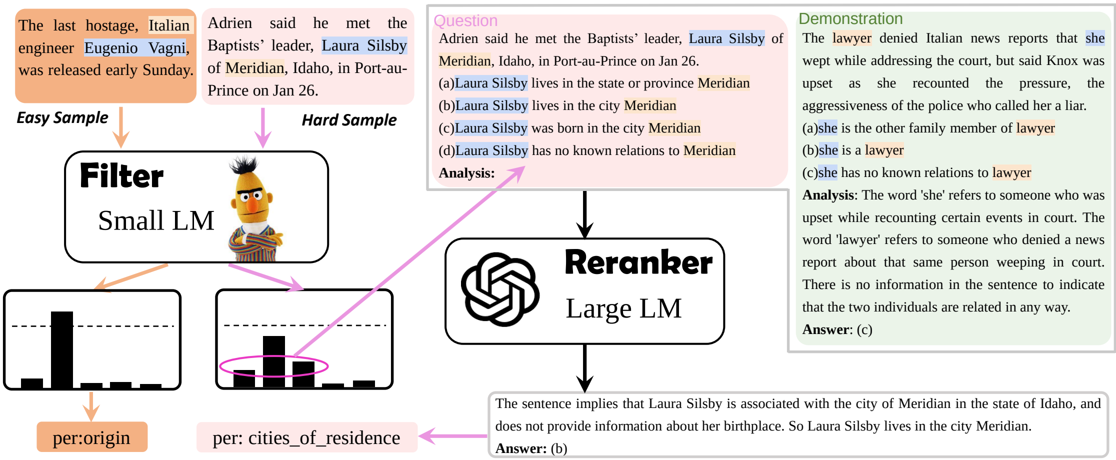


FIGURE 5.6: The architecture of our adaptive *filter-then-rerank* paradigm. We color easy samples in orange and hard samples in pink. For easy samples, the final predictions are exactly from the SLM-based methods. For hard samples, the top-N predictions from SLMs are fed into LLMs as the format of multiple-choice questions (pink box). The question is paired with demos (green box). LLMs rerank these N candidates and generate the final prediction.

5.4 Adaptive Filter-then-rerank Paradigm

Above findings can be summarized as: (1) SLMs generally outperform LLMs, especially with more training samples and fine-grained labels. (2) SLMs are much more time- and cost-efficient. (3) LLMs serve as powerful rerankers on *hard* samples that challenge SLMs. Based on them, we propose a simple, efficient, and effective adaptive reranker that combines the strengths of SLMs and LLMs.

5.4.1 Method

Our *adaptive filter-then-rerank* approach, shown in Figure 5.6, uses supervised SLMs as a filter to make preliminary decisions. Samples with confidence scores exceeding threshold are viewed as easy samples otherwise hard ones. For easy samples, we retain SLM predictions as final results. For hard samples, top-N predictions from SLMs are reranked via LLMs using ICL. Here LLMs employ MCQ prompts (Figure 5.4), containing demos and a sample to be reranked. The LLMs then generate the final answer and optionally provide an explanation.

5.4.2 Experimental Setup

We conduct experiments on FewNERD for NER task, TACREV for RE task and ACE05 for ED task. We employ top-performing SLM-based methods from Section 5.2 (FSLS or KnowPrompt) as the filter, and Vicuna-13B, InstructGPT or GPT-4 as the reranker. The threshold τ to determine sample difficulty is optimized on the valid set. For hard sample, the top-3 SLM predictions and `None` (if not included) are feed to LLMs for reranking. Each LLM prompt has 4-shot demos. We follow templates in Lu et al. [62] for TACREV and carefully design others. We adopt chain-of-thought reasoning [234], *i.e.,* prefacing the answer with an explanation, to facilitate LLMs' reranking procedure.

Baseline We compare our method with two kinds of baselines to validate its effectiveness. **(1) LLMs with ICL**: We follow the prompts in Section 5.2.3 and conduct experiments on three LLMs. **(2) Supervised SLMs**: We follow previous SoTA methods shown in Section 5.2.4 (FSLS or KnowPrompt). We additionally combine two SLMs with ensemble or reranking approach (*i.e.,* replace the LLM with another SLM as the reranker) to verify that improvements from our SLM-LLM integrated system are not solely due to the ensemble effects.

Table 5.3: Overall results of LLM-based ICL methods, SLM-based supervised methods, and our proposed *filter-then-rerank* (SLM+LLM) methods. The best results are in bold face and the second best are underlined. All results except InstructGPT and GPT-4 are averaged over 5 runs, and sample standard deviations are in the round bracket.

		FewNERD (NER)			TACREV (RE)			ACE (ED)		
		5-shot	10-shot	20-shot	20-shot	50-shot	100-shot	5-shot	10-shot	20-shot
LLM	CODEX	53.8(0.5)	54.0(1.4)	55.9(0.5)	59.1(1.4)	60.3(2.4)	62.4(2.6)	47.1(1.2)	47.7(2.8)	47.9(0.5)
	InstructGPT	53.6(–)	54.6(–)	57.2(–)	60.1(–)	58.3(–)	62.7(–)	52.9(–)	52.1(–)	49.3(–)
	GPT-4	-	-	57.8(–)	-	-	59.3(–)	-	-	52.1(–)
SLM	Previous SoTA	59.4(1.5)	61.4(0.8)	61.9(1.2)	62.4(3.8)	68.5(1.6)	72.6(1.5)	55.1(4.6)	63.9(0.8)	65.8(2.0)
	+ Ensemble (S)	59.6(1.7)	61.8(1.2)	62.6(1.0)	64.9(1.5)	71.9(2.2)	74.1(1.7)	56.9(4.7)	64.2(2.1)	66.5(1.7)
	+ Rerank (S)	59.4(1.5)	61.0(1.7)	61.5(1.7)	64.2(2.3)	70.8(2.3)	74.3(2.2)	56.1(0.3)	64.0(1.0)	66.7(1.7)
SLM + LLM	**Vicuna-13B**									
	+ Rerank (L)	60.0(1.8)	61.9(2.1)	62.2(1.4)	65.2(1.4)	70.8(1.6)	73.8(1.7)	56.9(4.0)	63.5(2.7)	66.0(2.6)
	+ Ensemble (S) + Rerank (L)	59.9(0.7)	62.1(0.7)	62.8(1.1)	66.5(0.5)	73.6(1.4)	75.0(1.5)	57.9(5.2)	64.4(1.2)	66.2(2.4)
	InstructGPT									
	+ Rerank (L)	60.6(2.1)	62.7(0.8)	63.3(0.6)	66.8(2.6)	72.3(1.4)	75.4(1.5)	57.8(4.6)	65.3(1.7)	67.3(2.2)
	+ Ensemble (S) + Rerank (L)	**61.3**(1.9)	**63.2**(0.9)	**63.7**(1.8)	**68.9**(1.3)	**74.8**(1.3)	**76.8**(1.2)	59.5(3.7)	65.3(1.9)	67.8(2.1)
	GPT-4									
	+ Rerank (L)	60.8(2.3)	62.6 (2.7)	63.0(1.3)	65.9(2.7)	72.3(0.3)	74.5(1.5)	59.6(2.9)	64.9(2.5)	67.1(2.5)
	+ Ensemble (S) + Rerank (L)	61.1(2.2)	62.8(0.9)	63.6(1.2)	68.6(1.3)	73.9(1.4)	75.9(2.4)	**60.9**(3.9)	**65.6**(1.5)	**67.8**(1.7)

5.4.3 Main Results

Few-shot Settings. Table 5.3 shows that our *filter-then-rerank* method consistently improves performance across three datasets and nine settings. For instance, with InstructGPT, reranking provides an average F1 gain of 2.4% without SLM ensemble (Lines 4 vs. 7). Based on ensemble SLMs as the filter, our method still achieves 2.1% (Lines 5 vs. 8) gains on average. This confirms (1) the effectiveness of the LLM reranking and (2) its gains are different and (almost) orthogonal to the SLM ensemble.

Large-scale Settings. Figure 5.7 demonstrates that our method impressively reduces budget and latency by approximately 80%˜90% compared to direct ICL. This reduction is due to (1) fewer LLM callings (only for hard samples) and (2) shorter prompts (fewer candidate labels and demos).

5.4.4 Analysis

Few makes big difference Our method selectively reranks hard samples. Table 5.4 shows that (1) only a minor fraction (0.5%˜10%) of samples are deemed

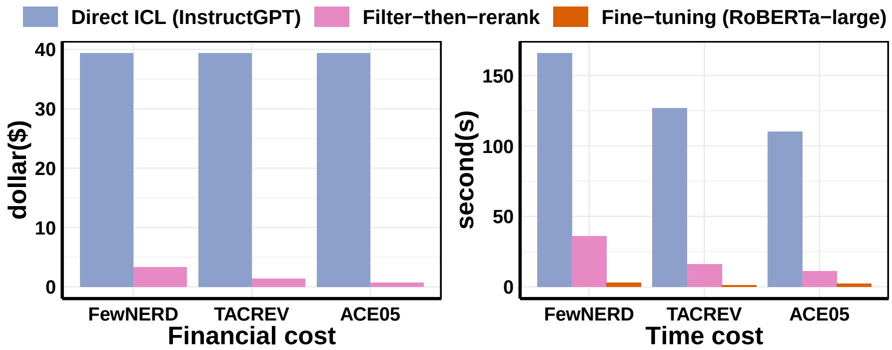


FIGURE 5.7: The financial and time cost over 500 sentences. InstructGPT as the reranker.

hard and are reranked by LLMs. (2) Despite their limited quantity, reranking results in a substantial performance boost on these samples (10%˜25% absolute F1 gains). This uplift on a small subset significantly enhances the overall performance.

TABLE 5.4: The F1-score differences before and after reranking on the reranked samples, as well as their proportion of the total samples.

	GPT-4				**InstructGPT**			
	before	after	△	ratio	before	after	△	ratio
FewNER	31.9	40.7	8.8	3.2%	31.4	28.3	−3.1	3.3%
TACREV	25.3	43.0	17.7	9.1%	33.8	43.4	9.6	7.1%
ACE05	31.1	57.9	26.8	1.6%	35.6	55.7	20.1	0.5%

GPT-4 is more aggressive From Tables 5.3 and 5.4, GPT-4 generally improves more on hard samples, yet InstructGPT surpasses GPT-4 in NER and RE tasks when evaluated overall. This discrepancy arises from GPT-4's aggressive reranking which introduces more true positives. InstructGPT, however, focuses more on reducing false positives.

5.4.5 Ablation Study

We investigate the effectiveness of the modules in adaptive *filter-then-rerank* system by removing each of them in turn: (1) **CoT**: We exclude the explantion for each examples in demo. (2) **Demo**: We remove all examples, rendering the reranking a zero-shot problem. (3) **LF** (label filtering): We retain all labels as candidate choices for reranking, instead of only the top-N labels from the SLMs. (4) **AD** (adaptive): We feed all samples, not just hard ones, to the LLMs.

TABLE 5.5: Ablation study on three datasets. The filter is ensembled SLMs and the reranker is GPT-4.

CoT	Demo	LF	AD	FewNERD (20-shot)	TACREV (100-shot)	ACE05 (20-shot)
✓	✓	✓	✓	**63.6**(1.2)	**75.9**(2.4)	**67.8**(1.7)
✗	✓	✓	✓	63.2(1.2)	75.4(2.4)	67.2(1.7)
✗	✗	✓	✓	63.0(1.4)	74.9(2.2)	66.6(1.5)
✗	✗	✗	✓	62.4(2.1)	73.8(2.5)	66.5(1.3)
✗	✗	✗	✗	12.5(2.7)	59.9(6.0)	5.4(1.1)
Previous SoTA methods				62.6(1.0)	74.1(1.7)	66.5(1.7)

We show their results in Table 5.5 and see that (1) Demos with explanations consistently enhance the reranking ability of LLMs across all datasets. (2) Demos without explanations also contribute to performance improvement. (3) Label filtering results in gains and notably reduces the demo length, hence cutting inference costs. (4) The performance collapses without a filter to identify sample difficulty, reiterating the need for an integrated SLM-LLM system to complement each other.

5.4.6 Case Study

Table 5.6 showcases some *hard* examples which benefits from our LLM reranking. In accordance with our intuition, we observe that the LLM rerankers correct two kinds of erroneous predictions made by LLMs. (1) The lack of external knowledge, such as the first (*Triptolemus is a figure in Greek mythology*) and third examples (*Minas Gerais is a state instead of city*). (2) Limited reasoning abilities, such as the second (*His wife's children are his children*) and the fourth (*The word "fought" in this sentence does not involve any physical violence*) examples.

5.5 Conclusion

Through an extensive empirical study on nine datasets spanning four IE tasks, we find that LLMs, despite their superiority in extreme low-resource scenarios, are not effective few-shot and efficient large-scale information extractors in general. They struggle with IE-related prompts, have limited demonstration capacity, and incur

TABLE 5.6: Examples of the samples corrected by LLM reranking. We sample four examples from NER, RE and ED tasks, respectively. **Sentences**: The sentences in which samples locate. We color the samples (entities or trigger words) to be identified. **Before**: The prediction before LLM reranking. Based on SLM-based methods. **After**: The reranked prediction using LLMs. **Rationales**: LLM-generated Explanations.

Sentences (entity/event)	Before	After	Rationales
		Named Entity Recognition	
According to it, King Eleusis and Cothonea (Cyntinia), parents of Triptolemus, are visited by Demeter, who rears their son, feeding him divine milk by day and placing him into the fire at night, which makes Triptolemus grow faster than mortal children normally do.	person-other ✗	other-god ✓	According to the context, Triptolemus is a figure in Greek mythology, the son of King Eleusis and Cothonea (Cyntinia). He was a demigod who was sent by Demeter to teach mankind the art of agriculture. So Triptolemus is a god in some legend/religious story.
		Relation Extraction	
Survivors include his wife, Sandra; four sons, Jeff, James, Douglas and Harris; a daughter, Leslie; his mother, Sally; and two brothers, Guy and Paul.	None ✗	per:children ✓	The sentence states 'Survivors include his wife, Sandra; four sons, Jeff, James, Douglas and Harris; a daughter, Leslie; his mother, Sally; and two brothers, Guy and Paul.' This indicates 'his' is the father of Jeff, James, Douglas and Harris, and the husband of Sandra. So his is the parent of Jeff.
Eliza Samudio, Bruno 's ex-girlfriend and his 4-month-old baby boy's mother, disappeared from her hometown of Belo Horizonte, Minas Gerais.	per:cities_of_residence ✗	per:states_of_residence ✓	The sentence states that 'Eliza Samudio, Bruno's ex-girlfriend and his 4-month-old baby boy's mother, disappeared from her hometown of Belo Horizonte, Minas Gerais.' This indicates that Eliza Samudio is from the city of Belo Horizonte, which is located in the state of Minas Gerais. So Eliza Samudio lives in the state or province Minas Gerais.
		Event Detection	
We have fought every institution, every law firm ... it 's cost us a fortune but now we 're going on a massive expansion.	Conflict.Attack ✗	None ✓	In the given sentence, the word fought is used to describe the action of the speaker fighting against various institutions and law firms. This does not involve any physical violence or court proceedings, so the word fought does not trigger any known event.

high inference costs. However, LLMs significantly improve the performance on *hard* samples when combined with SLM. Building on these insights, we propose an adaptive *filter-then-rerank* paradigm to leverage the strengths of SLMs and LLMs and mitigate their limitations. This approach consistently achieves promising results, with an average 2.4% F1 gain across multiple few-shot IE tasks, while minimizing latency and budget costs. In summary, *filter-then-rerank* paradigm is the first work (to our best knowledge) which achieves promising results under few-shot and large-scale IE tasks.

Part II

Document Understanding as Reading Comprehension

Chapter 6

Benchmarking Long-context Document Understanding

In the second part of this thesis (Chapter 6-Chapter 7), we shift focus to another key aspect of automatic document understanding, *i.e.,* document reading comprehension. Addressing the **multi-modal** challenge identified in Chapter 1.3, we examine native documents that combine text and visual elements in formats like PDFs and PowerPoints in the following two chapters.

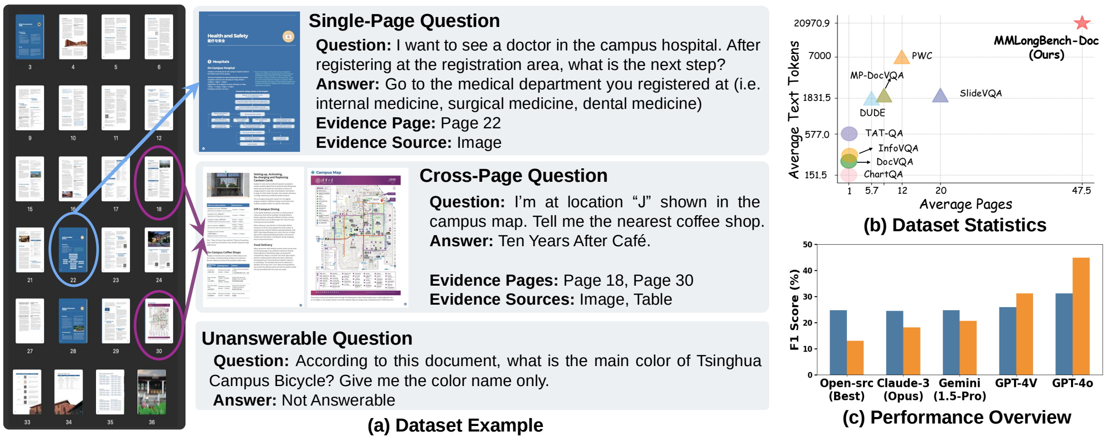


Figure 6.1: MMLongBench-Doc evaluates understanding abilities of LVLMs on lengthy documents that span tens of pages and incorporate multi-modal elements. Experiments (bottom-right) indicate that most LVLMs struggle, even falling behind LLMs that are fed with only OCR-parsed documents.

Specifically, this chapter [1] focuses on exploring the end-to-end generation capabilities of Large Vision-Language Models (LVLMs) for **long-context**, **multi-modal** reading comprehension tasks. Due to the absence of corresponding evaluation datasets, we introduce a novel benchmark designed for **long-context**, **multi-modal** scenarios to bridge the gap in this emerging field. Below, we present its motivation, construction process, evaluation results and analysis.

6.1 Introduction

Document reading comprehension is a long-standing task in urgent and practical needs. And the recent surge of LVLMs [44, 150, 235, 236] makes the end-to-end understanding on native multi-modal documents possible. Previous evaluations validate that most of them have achieved promising performance on single-page documents in DocVQA [24], ChartQA [152], InfoVQA [155], TAT-DQA [153], *etc.* However, considerable amounts of documents in the real world are long-context documents with tens or even hundreds of pages. The understanding of these lengthy documents brings new challenges for LVLMs from at least two aspects: (1) **Localization**: identify and retrieve information from massive, heterogeneous information (similar to the *needle in a haystack* task); (2) **Cross-page comprehension**: collect and reason over multi-source information across different pages. These two kinds of abilities are beyond the evaluation scopes of the aforementioned single-page documents. Some recent datasets [156–158] feature multiple-page reading comprehension, but almost all their documents are either as short of only several pages or of low information density, making the localization-related questions over-simple. Additionally, few (if any) questions in these datasets necessitate cross-page comprehension. In summary, there lacks a unified and high-quality benchmark on lengthy documents, leaving the evaluation of long-context reading comprehension largely unexplored.

In this paper, we present MMLongBench-Doc, a benchmark designed to evaluate the **M**ulti-**M**odality **Long**-context **Doc**ument reading comprehension abilities

[1]This chapter is published as Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. "MMLongBench-Doc: Benchmarking Long-context Document Understanding with Visualizations". 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks, pages 95963-96010, Vancouver, Canada, 2024. [37]

of LVLMs. Towards a comprehensive benchmark, it incorporates lengthy documents from both four existing datasets [152, 157–159] and other various papers, brochures, *etc.* Consequently, our benchmark includes 135 PDF-formatted documents spanning across 7 diverse domains, with each document averaging 47.5 pages and 21,214.1 textual tokens. Regarding the questions, we employ ten expert-level annotators to (1) edit questions associated with documents from existing datasets to meet our benchmark's standard and (2) create new questions for all collected documents to expand the scale of the benchmark. Then a three-round, semi-automatic reviewing process ensures the benchmark's annotation quality. As a result, MMLongBench-Doc comprises 1,082 human-annotated questions, with 184 sourced from four existing datasets and 898 newly annotated. Being a multi-modal benchmark, the answer to each question requires evidence from one or more of these five in-document sources: *text*, *layout*, *chart*, *table*, and *image*. Questions are categorized into three types based on the number of evidence pages [2], with examples illustrated in Figure 6.1(a): (1) 494 *single-page* questions (with one evidence page) mainly to evaluate localization abilities, (2) 365 *cross-page* questions (with multiple evidence pages) to assess cross-page comprehension, and (3) 223 *unanswerable* questions (no evidence for answering it, *i.e.,* no evidence pages) to reduce shortcuts and measure LVLMs' potential hallucinations. Meta-information including evidence pages, sources, and answer formats, is preserved for fine-grained evaluation and analysis. Detailed descriptions of the annotation pipeline and statistics can be found in Section 6.2.

We conduct extensive experiments on MMLongBench-Doc to evaluate the long-context reading comprehension abilities of 14 LVLMs, including 4 proprietary and 10 open-source ones. Given a document, we screenshot each page and feed all of these PNG-formatted images to LVLMs in an end-to-end approach. For comparison, we also convert the documents to textual format by OCR and evaluate another 6 proprietary and 4 open-source 10 LLMs (6 proprietary and 4 open-source ones). The results in Figure 6.1(c) highlight the challenges that current LVLMs face with long-context reading comprehension. The best-performing LVLM, GPT-4o, achieves an overall F1 score of only 44.9%, while the second-best LVLM, GPT-4V, scores 30.5%. Moreover, all the remaining LVLMs tested with multi-modal documents performed worse than single-modal LLMs handling lossy, OCR-parsed texts.

[2] Given a document D and a question q upon D, We call page P (in document D) an *evidence page* of q if the answer of q necessitates one or more pieces of evidence in page P.

Specifically, the Gemini-1.5-Pro and Claude-3-Opus present 4.2% and 6.4% absolute decrease when the inputs change from document screenshots to OCR-parsed texts. Regarding open-source models, the best-performing LVLM lags behind the best-performing LLM by 11.7%. These results reveal that long-context reading comprehension is a far-from-resolved task for current LVLMs.

6.2 MMLongBench-Doc Construction

We design a three-stage annotation pipeline for the construction of our benchmark. The three stages will be introduced in Section 6.2.1, Section 6.2.2, and Section 6.2.3, respectively. We also provide key statistics of our benchmark in Section 6.2.4.

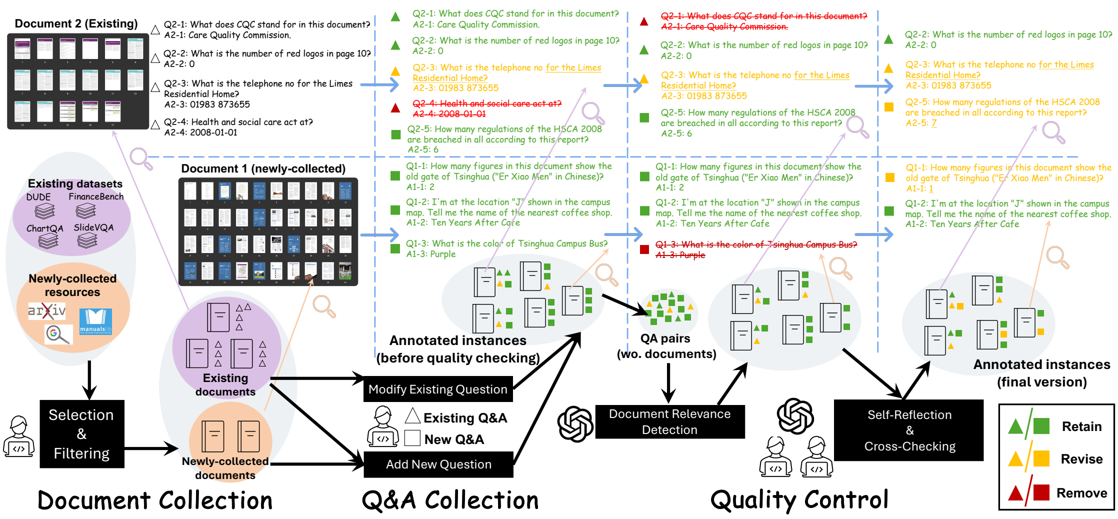


Figure 6.2: The annotation pipeline of MMLongBench-Doc.

6.2.1 Document Collection

As a long-context reading comprehension benchmark, the documents shall be of diverse topics and lengthy enough. To this end, we crawl a great amount of documents from various sources. Then we select the lengthy ones from these documents. Specifically, we encompass a diverse array of documents from two approaches. (1) **Existing documents** from four previous datasets: DUDE [157], SlideVQA [158], ChartQA [152], and FinanceBench [159]. (2) **Newly-collected documents** from

Arxiv [3], ManualsLib [4] and Google Search [5]. Then we (1) filter out the documents with fewer than 15 pages or license restrictions and (2) down-sample documents from DUDE, SlideVQA, and FinanceBench for a more balanced distribution.

In summary, we collect a total of 135 documents. Among them, 76 documents are from existing datasets (colored in purple and exemplified as Document 2 in Figure 6.2) and incorporate previously annotated questions (represented as triangles). The remaining 59 documents are newly collected (colored in orange and exemplified as Document 1) and incorporate no existing questions. We manually categorize them into 7 types: *Research Report*, *Financial Report*, *Academic Paper*, *Brochure*, *Guideline*, *Administration & Industry File*, *Tutorial / Workshop*.

6.2.2 Question and Answer Collection

To serve as a high-quality and comprehensive benchmark, the question annotation of our benchmark adheres to the following standards: (1) All questions shall be neither over-easy nor over-difficult. (2) Questions are not repetitively derived from the same page or the same pattern. (3) The distribution of evidence numbers, evidence sources, and evidence locations for the questions shall be balanced. (4) No questions shall be answered correctly without accessing the relevant documents.

Ten authors serve as expert-level annotators for the question-and-answer collection. All of them are doctors or Ph.D. students proficient in English reading and writing. Before formal annotation, they undergo a training session and pre-annotate three documents for practice. We iteratively review their annotation results and provide personalized feedback until their annotations meet the standards mentioned above. Regarding the formal annotation, we divide 135 documents into 54 batches (each having 2-4 documents) and dispatch these batches to annotators. We then ask the annotators to submit their results in units of batches and set reasonable time intervals for each batch's submission. We timely evaluate their annotations after each submission and remind the annotators if their questions in this turn diverge from the standards. It avoids the annotators rushing all assignments in a short time and benefits the annotation quality. We recommend the annotators take

[3] `https://arxiv.org`
[4] `https://www.manualslib.com`
[5] `https://www.google.com.sg`

60-90 minutes on each document. Specifically, the annotators shall rapidly read through the whole document in the first 15-30 minutes. For the remaining time, they shall dive deep into specific components to modify existing annotations and/or add new annotations as detailed below.

Modify Existing Questions. Documents collected from existing datasets had been annotated with some questions and answers from previous work. However, their crowd-sourcing annotations inevitably make some questions, answers, and other meta information unqualified. Therefore, we edit their annotations before including them as a component of our benchmark.

Specifically, we classify six potential problems in original annotations. They are *Wrong Answers or Evidence Pages*, *Repetitive Question*, *Ambiguous Question*, *Decontextualization-required Question*, *Low Document-relevant Question* and *Potential Shortcut*. Given an existing document, the annotators are tasked to evaluate each existing question's quality according to whether they have one or more above problems and assign a label from {`Retain`, `Revise`, `Remove`} for each question. Then the annotators would revise the `Revise` questions to meet our quality criteria and remove the `Remove` questions. See triangle-marked questions colored in green, yellow, and red in Figure 6.2 as `Retain`, `Revise`, `Remove` examples. Among all 425 original questions from 76 existing documents, 32.2% of them are revised and 46.1% are removed. We finally collect 211 questions in this procedure.

Add New Questions. We newly annotate questions on both existing and newly collected documents to expand the questions in our benchmark. See square-marked questions in Figure 6.2 as examples. Specifically, we ask annotators to add about 3 questions on existing documents, and 6 questions on newly-collected documents. Given most existing questions are single-page ones and sourced from texts, we put more focus on (1) cross-page and unanswerable questions and (2) questions sourced from tables, charts, and images for newly added questions to balance the distribution. Associated with questions, annotators also provide reference answers and meta-information (*i.e.,* evidence sources, answer format, evidence locations) for all samples. We finalized a collection of 965 samples in this procedure.

6.2.3 Quality Control

Combining the merits of humans and LVLMs, we adopt a three-round, semi-automatic quality control procedure to improve the annotation quality of our benchmark. We detail each round in the following components.

Document-relevant Detection. Our benchmark is designed to evaluate LVLMs' long-context document understanding abilities. All questions are expected to be unanswerable without access to corresponding documents. To remove low document-relevant questions (*i.e.,* questions not relying on documents), we feed each annotated question **WITHOUT** documents to GPT-4o. A question will be identified as *low document-relevant* question if GPT-4o correctly predicts under this case. Ultimately, 94 samples are identified as low document-relevant questions and removed in this round.

Self-reflection. We draw inspirations from MMBench [137] and leverage LVLMs to reduce the wrongly-annotated samples. Specifically, we feed the remaining questions from the last round **WITH** their documents to GPT-4o. Samples whose model predictions are inconsistent with the reference answers are sent back to corresponding annotators. The annotators are asked to check each question and identify whether the inconsistency is caused by *problematic annotation* or not. As a result, 13.8% of the samples are identified as problematic annotations. The annotators revise them accordingly.

Cross-checking. In parallel, annotators cross-check the annotated samples from other annotators and determine the inconsistency reasons the same as described above. We calculate Cohen's kappa value of their identifications as 0.42 (17.5% inconsistent samples), showing a moderate agreement. Regarding the 17.5% inconsistent samples, two primary authors serve as meta-annotators and make final decisions on them (and if necessary, revise accordingly).

6.2.4 Dataset Overview and Analysis

The main statistics of MMLongBench-Doc are presented in Table 6.1. Overall, our benchmark consists of 1,082 questions. These questions are constructed upon 135 lengthy documents across 7 document types, with an average of 47.5 pages and

Statistic	Number
Documents	135
- Type	7
- Average/Medium pages	47.5 / 28
- Average/Medium length	21,214.1 / 12,179
Total questions	1,082
- Single-page question	494 (45.7%)
- Cross-page questions	365 (33.7%)
- Unanswerable questions	223 (20.6%)
- Derived questions	184 (17.0%)
- Newly-annotated questions	898 (83.0%)
(Evidence source)	
- Pure-text	305 (35.5%)
- Layout	119 (13.9%)
- Table	218 (25.4%)
- Chart	178 (20.7%)
- Image	304 (35.4%)
(Answer Format)	
- String	250 (29.1%)
- Integer	299 (34.8%)
- Float	159 (18.5%)
- List	151 (17.6%)
Avg./Max. question length	16.4 / 60
Avg./Max. answer length	2.8 / 54

TABLE 6.1: Dataset Statistics

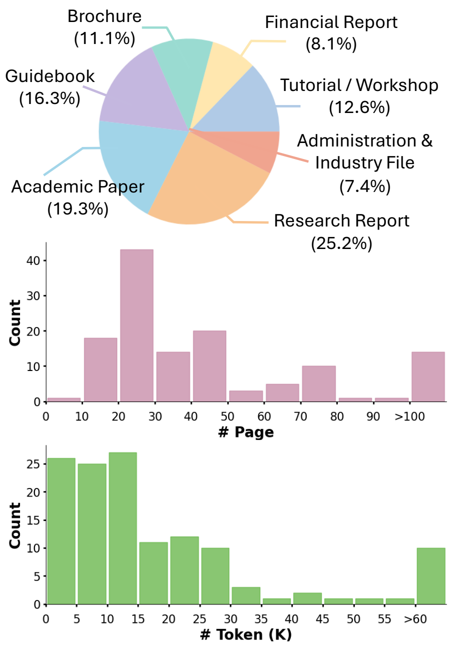


FIGURE 6.3: Detailed distribution of documents. **Top**: Document type. **Middle**: Page Number. **Bottom**: Token Number.

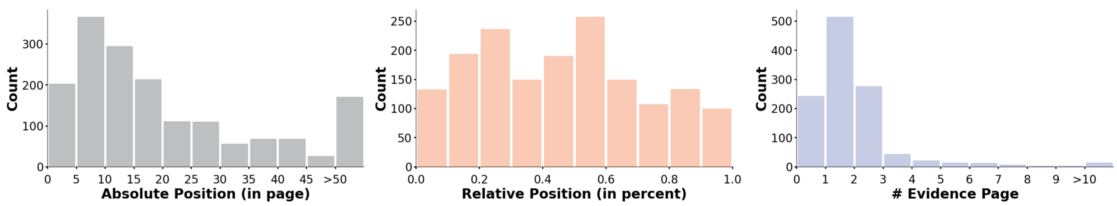


FIGURE 6.4: Detailed distribution of questions & answers. **Left**: Absolute position of answer evidences (the page index). **Middle**: Relative position (the page index/document page number). **Right**: Evidence page number of each question. (0: unanswerable question; ≥ 2: cross-page question).

21,214.1 tokens. Please see detailed distributions of these documents in Figure 6.3. Regarding the questions, there are 494 single-page questions (1 evidence page), 365 cross-page questions (2+ evidence pages), and 223 unanswerable questions (no evidence page). These three types of questions evaluate the LVLMs's long-context reading comprehension capabilities from complementary aspects: the localization ability, the cross-page comprehension ability, and the hallucination severity, respectively. For single-page and cross-page questions, their answer evidence is scattered among different context sources (*i.e.*, text, layout, table, chart, image) and evenly distributed across different locations of the documents (see Table 6.1, Figure 6.4 Left and Middle). Also notably, 28.6% of cross-page questions have more than two evidence pages, which further enhances the challenge of our benchmark.

6.3 Evaluation

6.3.1 Evaluation Protocol

We follow MATHVISTA [237] to conduct a three-step evaluation protocol: *response generation*, *answer extraction*, and *score calculation*. We adopt such a protocol out of three considerations: (1) Current LVLMs are instructed to generate long responses, rather than short-form answers, in conventional settings. (2) The evaluation of long responses, however, remains an open and challenging problem. (3) We focus on the document understanding (not instruction following) abilities of LVLMs. Specifically, we impose no limitations on *response generation* stage to encourage LVLMs to answer the questions in a freestyle. Then we propose a unified LLM-based *answer extractor* (GPT-4o under our setting) to convert their long responses to short-form answers. Finally, we use a rule-based *score calculator* to evaluate the converted short answers. We report both generalized accuracy and generalized F1 score to balance the answerable (positive) and unanswerable (negative) questions.

6.3.2 Experimental Setup

We evaluate 14 LVLMs on MMLongBench-Doc, including 4 proprietary LVLMs and 10 open-source LVLMs. To purely evaluate LVLMs' long-context reading comprehension abilities, we screenshot each page of the PDF-formatted document with 144 DPI and feed all these PNG-formatted images to LVLMs in an end-to-end approach. Notably, all evaluated open-source LVLMs do not support multi-image inputs or present significant performance drops when fed with excessive images (*e.g.,* more than 10 or 20 images). Therefore, we employ a concatenation strategy that combines all screenshot pages into 1 or 5 images and feeds these concatenated images to open-source LVLMs. Regarding proprietary LVLMs, we adopt the same concatenation strategy and reduce the image number to 20 for Claude-3-Opus to fit its maximum image threshold. For GPT-4o, GPT-4V, and Gemini-1.5-Pro, we directly send all original screenshots to them (*i.e.,* the image number equals the page number).

For comparison, we also use the Tesseract [238] OCR model to recognize and extract texts from the documents and feed the parsed documents to 10 LLMs, including 6 proprietary and 4 open-source ones. Texts exceeding their context lengths are truncated. Notably, as a key component of the classical solution for the reading comprehension task, the OCR model can handle most flattened texts and some structured tables in the document. However, it cannot perceive the information from the charts or images. Thus the TXT-formatted, OCR-parsed documents are lossy documents in which the information is not fully preserved. Additionally, we also conduct manual evaluation on a subset of our datasets (238 questions from 29 documents) to indicate the difficulty of this task for humans.

6.3.3 Main Results

We compare the performance of different LVLMs and LLMs in Table 7.2, reporting their generalized accuracy and F1 scores (shown in the last two columns). Regarding LVLMs, we draw several conclusions as below: (1) The performance demonstrates that long-context reading comprehension is still a challenging and unsolved task for current LVLMs. The best-performing LVLM, GPT-4o, merely achieves a 44.9% F1 score. The second best-performing LVLM, GPT-4V, lags behind by over 10% percent and presents a 31.4% F1 score. All other LVLMs only achieve about 20% or even lower F1 scores. (2) Though far from satisfactory, GPT-4o performs much better than all other models (including GPT-4V). Thus we speculate that the multi-modal pre-training paradigm significantly benefits LVLMs' cross-modality understanding capabilities. (3) Proprietary LVLMs perform better than open-source LVLMs by a large margin. We attribute it to the difference of acceptable image numbers: open-source LVLMs only support single-image or several-image inputs, while proprietary LVLMs can be fed with at least 20 images or even more. Given that lengthy documents have tens of even hundreds of pages, it is impractical for open-source LVLMs to accurately perceive the information in the documents from the excessively concatenated images. (4) The performances of different models are highly correlated with their acceptable image

TABLE 6.2: **Evaluation results on MMLongBench-Doc.** We report the generalized accuracy of five types of evidence sources including pure text (TXT), layout (LAY), chart (CHA), table (TAB), and image (IMG). We also present the generalized accuracy of questions categorized by the number of evidence pages: single-page (SIN), cross-page (MUL), and unanswerable (UNA) questions. The **best** and **second-best** performance in each section are highlighted.

Model	#Param	Context Window	Evidence Source					Evidence Page			ACC	F1
			TXT	LAY	CHA	TAB	FIG	SIN	MUL	UNA		
OCR (Tesseract [238]) + Large Language Models (LLMs)												
Open-source Models												
ChatGLM-128k [167]	6B	128k	23.4	12.7	9.7	10.2	12.2	18.8	11.5	18.1	16.3	14.9
Mistral-Instruct-v0.2 [239]	7B	32k	19.9	13.4	10.2	10.1	11.0	16.9	11.3	24.1	16.4	13.8
Mixtral-Instruct-v0.1 [240]	8x7B	32k	24.2	14.8	12.5	15.0	13.7	21.3	14.1	13.1	17.0	16.9
Mixtral-Instruct-v0.1 [240]	8x22B	64k	34.2	21.3	19.5	21.3	19.2	27.7	21.9	32.4	26.9	24.7
Proprietary Models												
QWen-Plus [241]	-	32k	17.4	15.6	7.4	7.9	8.8	14.2	10.6	42.2	18.9	13.4
DeepSeek-V2 [242]	-	32k	27.8	19.6	8.8	17.0	9.4	20.2	15.4	48.1	24.9	19.6
Claude-3 Opus [243]	-	32k	30.8	30.1	16.4	24.4	16.3	32.0	18.6	30.9	26.9	24.5
Gemini-1.5-Pro [244]	-	32k	29.3	15.9	12.5	17.7	11.5	21.2	16.4	**73.4**	31.2	24.8
GPT-4-turbo [44]	-	128k	36.5	21.0	20.7	24.3	17.3	28.7	23.8	31.2	27.6	25.9
GPT-4o [245]	-	128k	41.1	23.4	28.5	38.1	22.4	35.4	29.3	18.6	30.1	30.5
Large Visual Language Models (LVLMs)												
Open-source, 7-14B Models												
DeepSeek-VL-Chat [246]	7.3B	4k	7.2	6.5	1.6	5.2	7.6	5.2	7.0	12.8	7.4	5.4
Idefics2 [247]	8B	8k	9.0	10.6	4.8	4.1	8.7	7.7	7.2	5.0	7.0	6.8
MiniCPM-Llama3-V2.5 [248, 249]	8B	2k	11.9	10.8	5.1	5.9	12.2	9.5	9.5	4.5	8.5	8.6
InternLM-XC2-4KHD [150]	8B	16k	9.9	14.3	7.7	6.3	13.0	12.6	7.6	9.6	10.3	9.8
mPLUG-DocOwl 1.5 [235]	8.1B	4k	8.2	8.4	2.0	3.4	9.9	7.4	6.4	6.2	6.9	6.3
Qwen-VL-Chat [250]	9.6B	6k	5.5	9.0	5.4	2.2	6.9	5.2	7.1	6.2	6.1	5.4
Monkey-Chat [251]	9.8B	2k	6.8	7.2	3.6	6.7	9.4	6.6	6.2	6.2	6.2	5.6
Open-source, ¿14B Models												
CogVLM2-LLaMA3-Chat [252]	19B	8k	3.7	2.7	6.0	3.2	6.9	3.9	5.3	3.7	4.4	4.0
InternVL-Chat-v1.5 [236]	26B	4k	14.0	16.2	7.1	10.1	16.6	14.9	12.2	17.5	14.6	13.0
EMU2-Chat [253]	37B	2k	6.1	9.7	2.6	3.8	7.7	5.7	6.1	16.5	8.3	5.5
Proprietary Models												
Claude-3 Opus [243]	-	200k	24.9	24.7	14.8	13.0	17.1	25.6	13.8	7.6	17.4	18.1
Gemini-1.5-Pro [244]	-	128k	21.0	17.6	6.9	14.5	15.2	21.1	11.1	69.2	28.2	20.6
GPT-4V(ision) [44]	-	128k	34.4	28.3	28.2	32.4	26.8	36.4	27.0	31.2	32.4	31.2
GPT-4o [245]	-	128k	**46.3**	**46.0**	**45.3**	**50.0**	**44.1**	**54.5**	**41.5**	20.2	**42.8**	**44.9**
Human Baseline												
Human Experts	-	-	-	-	-	-	-	-	-	-	65.8	66.0

numbers and maximum image resolutions. Notably, open-source LVLMs that support high-resolution images (*i.e.*, InternLM-XC2-4KHD and InternVL-Chat-v1.5) exhibit superior performance compared to those with lower resolution limits.

Surprisingly, LVLMs even demonstrate overall worse performance than LLMs, even LLMs are fed with lossy OCR-parsed documents. Specifically, Gemini-1.5-Pro and Claude-3 Opus have 4.2% and 6.4% absolute F1-score degradations on vision versions. And the best-performing LLM (Mixtral) also surpasses the best-performing LVLM (InternVL-v1.5) by 11.7%. The above results clearly reveal that most current LVLMs are still not proficient in cross-modality, long-context document understandings. It is promising that GPT-4o and GPT-4-turbo achieve better performance when seeing multi-modality PDF documents than parsed text by 14.4%

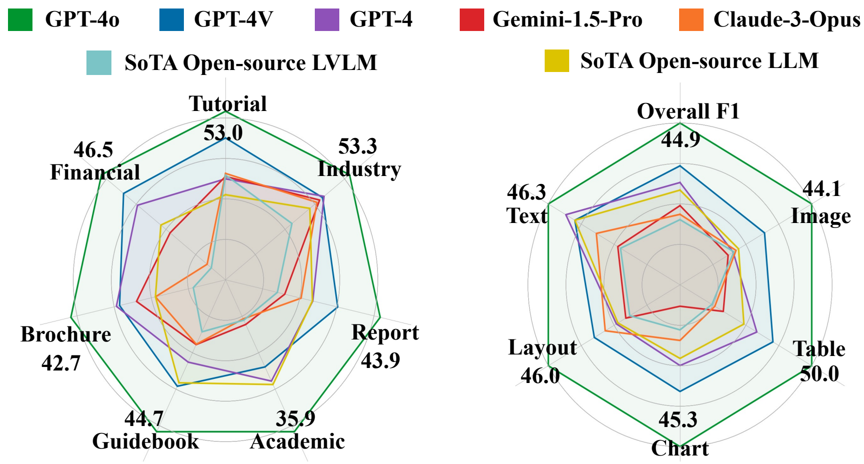


FIGURE 6.5: Fine-grained results on document types and evidence sources.

and 5.3% F1-score, respectively. Their performances validate the feasibility, benefit, and necessity of understanding documents in an end-to-end, cross-modality approach. We speculate that the scarce related pre-training corpus (*i.e.*, extremely multi-image or lengthy documents) hinders the long-context reading comprehension capabilities of other LVLMs. We leave related explorations for future work.

Regarding the human evaluation, we observe 66.0% F1-score from our annotators and a significant performance gap (exceeding 20% in absolute) between the current LVLMs and humans. This gap highlights the challenges of document understanding for LVLMs and the necessity of our benchmark.

6.3.4 Fine-grained Results.

Document Type. As illustrated in Figure 6.5, LVLMs and LLMs exhibit distinct performance patterns across various document types. Our findings include: (1) All evaluated models demonstrate decent performance on industrial documents, which tend to have more standardized formats and less non-textual information. (2) The GPT series and Mixtral (*i.e.*, the SoTA open-source LLM) show relatively balanced performance across different document types. In contrast, other models perform significantly worse in specialized domains such as academic papers and financial reports. (3) When equipped with OCR, LLM-based models like GPT-4 and Mixtral achieve comparable or even superior performance on industrial documents, academic papers, and brochures. Conversely, end-to-end LVLMs outperform OCR+LLMs in areas such as tutorials, research reports, and guidelines. We speculate that comprehending these latter document types requires more extensive multi-modal information, from which LVLMs significantly benefit.

Evidence Source. We categorize questions based on their evidence sources and present fine-grained results in Figure 6.5 and Table 7.2. Our observations reveal that only GPT-4o exhibits relatively balanced performance across the different sources. Other LVLMs, however, show inferior performance on questions related to charts and/or images compared to those related to text and/or layout. Additionally, LLMs generally demonstrate better or comparable performance to LVLMs on text- and table-related questions but show worse performance on questions involving other elements. This highlights the limitations of OCR (and other PDF parsers) when dealing with charts and images, as well as the gap in OCR capabilities between LVLMs and pure-text LLMs.

Evidence Position. We also examine how the evidence locations (*i.e.*, the page indexes where the answer evidence is found) affect model performance. The results shown in Figure 6.6 reinforce that MMLongBench-Doc poses significant challenges for current models, at least partially due to the extended length of the documents. Almost all models (except InternVL-v1.5) exhibit their best performance on questions derived from the initial pages, while their performance declines progressively as the page index increases. Interestingly, two proprietary models, Gemini-Pro-1.5 and Claude-3-Opus, experience particularly sharp declines in performance.

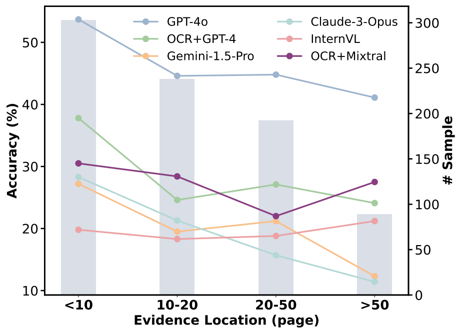


FIGURE 6.6: Relationships between evidence positions and model performances.

Number of Evidence Page. We observe a consistent trend that all models achieve higher scores on single-page questions than cross-page questions. It reveals that gathering and reasoning over all necessary information across different pages is not trivial for current LVLMs and LLMs. More interestingly, evaluated LVLMs behave differently on unanswerable questions. GPT-4o and Claude-3 Opus adopt more aggressive strategies and usually tend to provide some answers. It makes their answers more likely helpful, but also increases the risk of hallucination and unfaithfulness (see their scores on unanswerable questions are much lower than answerable questions). On the contrary, Gemini-1.5-Pro, DeepSeek-VL-Chat, and EMU2-Chat are much more cautious and tend to refuse to answer questions about which they are uncertain. It makes their answers safer but less helpful (with large amounts of responses like *I don't know*).

6.4 Analysis & Discussion

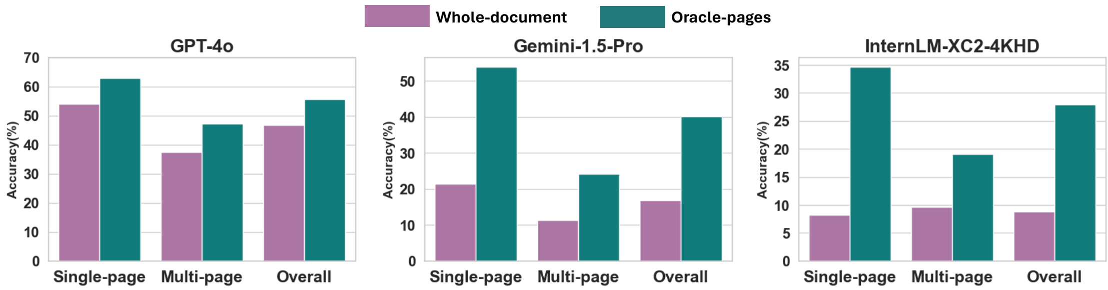


FIGURE 6.7: Performance comparisons between normal setting (feeding models with the whole documents) and oracle setting (feeding models only with the evidence pages) among three LVLMs.

6.4.1 Oracle Setting

We conduct additional experiments to explore to what extent the challenges of MMLongBench-Doc are caused by the long-context lengths of documents. Specifically, we feed 820 answerable questions along with their oracle evidence pages (instead of the whole documents) to three representative LVLMs and show results in Figure 6.7. On one hand, it indicates that long-context length is a significantly challenging factor for document understanding. Compared with the oracle-page setting, lengthy documents lead to more than 20% absolute performance degradation on Gemini-1.5-Pro and InternLM-XC2-4KHD. Regarding the single-page questions, the performance difference even achieves up to 30%. On the other hand, the overall performance achieves only about 40% and 30% for Gemini-1.5-Pro and InternLM-XC2-4KHD even under oracle-page setting. And the improvement for GPT-4o is much less (about 10%). It demonstrates that the development of long-context LVLMs can largely facilitate, though still can not fully solve, the long-context reading comprehension task.

6.4.2 Error Analysis

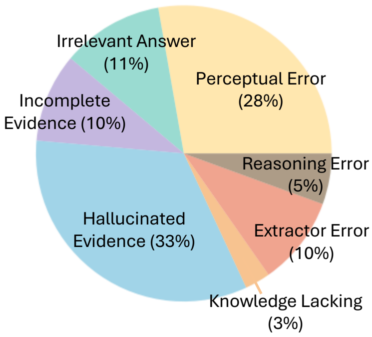


FIGURE 6.8: Error distribution

We further conduct error analysis to understand the bottleneck of current LVLMs in a qualitative approach. Specifically, we randomly select 72 error predictions from GPT-4o's responses and manually check their error reasons. These errors are categorized into seven types: *Perceptual Error*, *Irrelevant Answer*, *Incomplete Evidence*, *Hallucinated Evidence*, *Extractor Error*, *Reasoning Error* and *Knowledge Lacking*. The distribution of these errors is illustrated in Figure 6.8. It indicates that most errors come from the model's hallucination (*i.e.,* wrong explanations and answers to unanswerable questions) and perceptual errors (mainly in visual contexts). Additionally, GPT-4o sometimes misunderstands the intent of questions and provides irrelevant responses. The errors caused by collecting incomplete evidence (for cross-page questions) are also unignorable.

6.4.3 Case Study

In this section, we compare the responses from six representative models. We show one case in Figure 6.9 and observe that: (1) GPT-4o outperforms all other models by a significant margin. (2) While LLMs demonstrate performance comparable to LVLMs on text-related questions, they lag behind on questions concerning other vision-intensive sources such as charts, layouts, and images. (3) Gemini-Pro-1.5 tends to adopt a conservative approach, often opting not to provide an answer. (4) Even the best-performing open-source LVLM, InternVL-v1.5, struggles with these cases, indicating that open-source LVLMs still fall short in tasks involving long-context document understanding.

6.4.4 Limitations

While our benchmark addresses significant gaps in the previous datasets, we acknowledge several limitations.

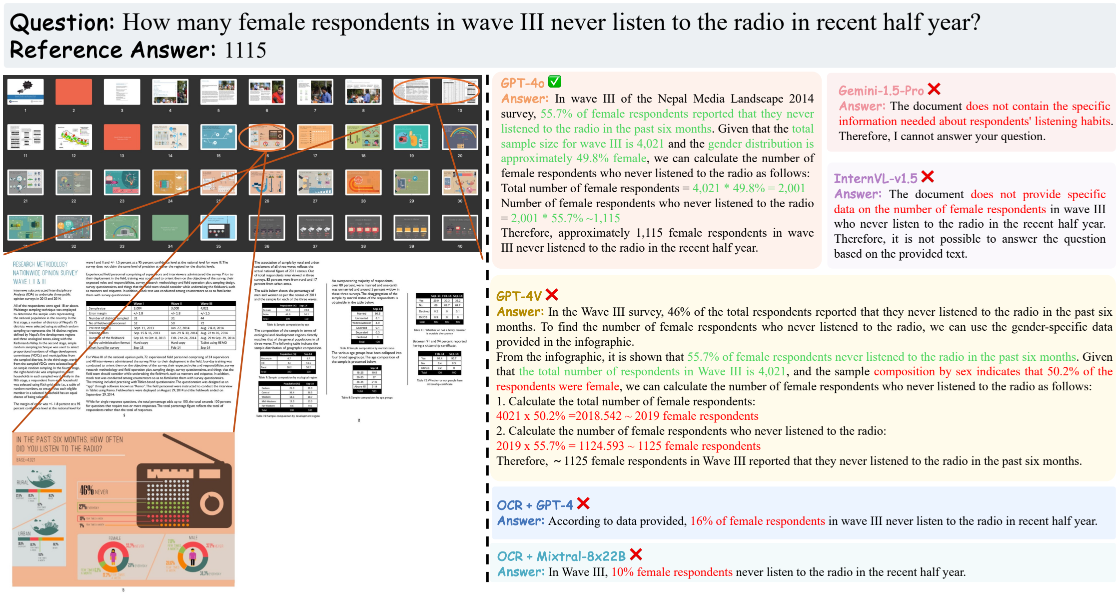


FIGURE 6.9: Case Study. Evidence sources: two charts and one table. The three evidence pages of this question are zoomed in. We manually analyze the responses step-by-step: the correct extracted information and reasoning are colored in green, and the wrong ones are colored in red.

One primary limitation is the scale of the benchmark. Currently, our benchmark includes a test set comprising 135 documents and 1,082 questions. It is much smaller compared to previous datasets. The complexity and difficulty of annotations limit the scale of our benchmark. As a long-context benchmark, our documents average about 50 pages and 20,000 tokens. And most questions require either complicated reasoning or cross-page comprehension. It takes more than one hour for an expert-level annotator to read through a single document, and then edit existing instances and create new instances on this document. Given the purpose of MMLongBench-Doc as an evaluation benchmark, we prioritize annotation quality over quantity. Moreover, the evaluation results confirm that the scale of our benchmark is sufficient for fine-grained evaluations across different document types, evidence sources, evidence pages, *etc.* Additionally, we plan to expand our benchmark by adding more documents and questions in future iterations.

We roughly categorize these questions into three types, *i.e.,* single-page, cross-page, and unanswerable questions, based on whether evidence can be found in the documents and the number of evidence pages. However, unlike MMBench [137] or MathVista [237], we provide no further taxonomy to classify some (*e.g.,* 7 or 20) fine-grained, evaluated reasoning or perception capabilities out of two main reasons: (1) Prior (*i.e.,* pre-annotation) taxonomy limits the diversity of the questions.

Therefore we provide no predefined classifications in our guideline and encourage the expert-level annotators to freely write questions without constraints. (2) The intrinsic complexity of document understanding presents significant challenges for establishing a posterior (*i.e.,* post-annotation) taxonomy.

While there exist limitations in our benchmark, MMLongBench-Doc surely represents a significant step forward in this field. We would iteratively maintain and refine this benchmark and hope it could push forward the development of long-context document understanding.

6.5 Conclusion

In this work, we present MMLongBench-Doc to evaluate the reading comprehension capabilities of LVLMs under multi-modal, long-context scenarios. Extensive experiments on 14 LVLMs (and 10 LLMs for comparison) reveal that the understanding of lengthy documents poses great challenges to current LVLMs. Even though the performance of GPT-4o proves the benefit of end-to-end, multi-modality perception on documents, most LVLMs struggle on long visual contexts (*i.e.,* extremely multiple images) and show inferior performance compared to OCR+LLM pipelines. We hope that the construction of our benchmark could push forward the development of more powerful LVLMs on lengthy document understanding.

Chapter 7

Empirical Study on Efficient Visualized Document Retriever

In the last chapter, we reveal that LVLMs still face challenges in processing multi-modal, long-context documents for end-to-end reading comprehension. By reducing document length, Retrieval-augmented Generation (RAG) can smoothly address this limitation. Additionally, RAG also enables the selective reading from a large document corpora.

As outlined in Section 2.2.2, Visualized Document Retrieval (VDR) is the core component of visual RAG systems. However, top-performing VDR models often suffer from high memory usage and latency. To mitigate this problem, this chapter [1] focuses on developing an efficient VDR strategy maintaining retrieval performance while minimizing resource requirements. We hope that this exploration advances the wide applications of VDR on **multi-modal**, **large-scale** document system.

7.1 Introduction

Visualized Document Retrieval (VDR) retrieves the most related image-formatted documents given user queries. Unlike conventional retrieval systems where raw

[1] This chapter is published as Yubo Ma, Jinsong Li, Yuhang Zang, Xiaobao Wu, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Haodong Duan, Jiaqi Wang, Yixin Cao, and Aixin Sun. "Towards Storage-Efficient Visual Document Retrieval: An Empirical Study on Reducing Patch-Level Embeddings". Findings of the Association for Computational Linguistics: ACL 2025, pages 19568–19580, Vienna, Austria, 2025. [40]

text must be parsed before indexing, VDR captures documents as images (screenshots) and encodes them into embeddings using LVLMs. By preserving layout structures and visual elements in documents, it enables retrieval in a what-you-see-is-what-you-get manner. As a result, VDR achieves superior retrieval accuracy and demonstrates strong potential across various applications [254, 255].

The state-of-the-art visualized document retriever, ColPali/ColQwen2 [38], represents a significant advancement in this field. As shown in Figure 7.1 (left), ColPali/ColQwen2 encodes each document page as N_p patch-level embeddings during the offline stage and saves them for online computation. While the excessive number of patch embeddings enables the perceiving of fine-grained details (which is particularly important for document-related images), it introduces substantial memory footprints and computational overhead in both offline indexing storage and online similarity computation. For example, a medium-sized document with 50 pages requires about 10 MB memory for embedding storage [2]. This substantial memory footprint presents a bottleneck for scalability and practical deployment of VDR systems under real-world scenarios.

In this work, we present an in-depth analysis of the storage-efficient visualized document retriever, exploring ***how to reduce each page's patch embedding number with minimal performance degradation***. We consider two common token-reduction approaches, *i.e.,* token pruning [178] and token merging [188], respectively. In terms of token pruning, we investigate multiple pruning strategies in Section 7.2 and aim to retain only the high-informative patch embeddings. Even though token pruning works to some extent, it can not reduce the embedding numbers by orders of magnitude without significant performance drops. More embarrassingly, we observe that a simple baseline, *i.e.,* random pruning, is competitive and performs better than most other carefully-designed strategies. With deeper analysis on this observation, we propose the hypothesis that (1) the informativeness of patch embedding is highly conditioned on the queries, which are unknown and unpredictable during the offline indexing stage. (2) the patch embeddings can be grouped and, accordingly, are prone to be dropped by the group under some specific criteria. The above two reasons make it impractical to decide which embeddings should be pruned without access to the queries. Therefore, we speculate that pruning-related strategies are inappropriate under VDR settings.

[2] ColQwen2 divides each page into 768 tokens, each represented by a 128-dimensional vector. Stored as 16-bit floats, it requires 50*768*128*16 bits = 9.6 MB per document.

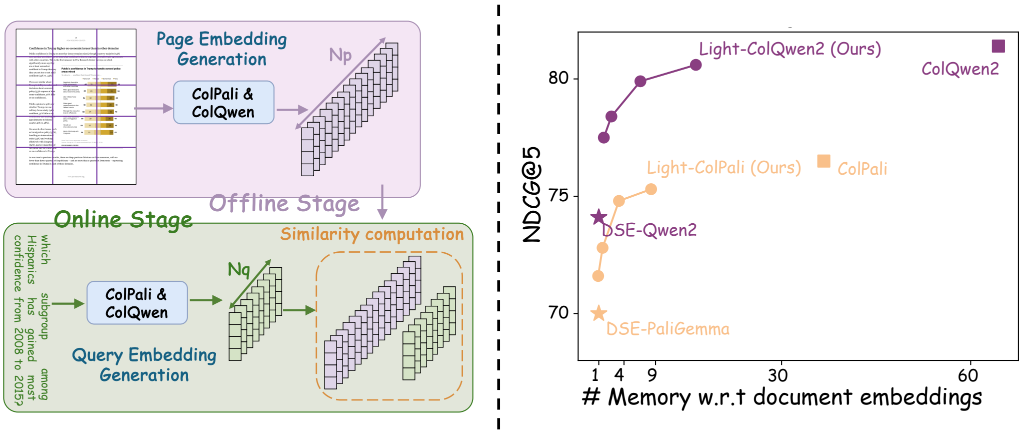


FIGURE 7.1: **Left**: The diagram of Visual Document Retriever equipped with ColPali/ColQwen2 retriever. It encodes each page into N_p patch-level embeddings and thus incurs prohibitive memory cost. **Right**: This work aims to reduce the saved embedding numbers at the minimum performance drop. Our simple yet effective approach, Light-ColPali/ColQwen2, retains most of the performance but with significantly reduced memory cost.

In Section 7.3, we turn into token merging approaches and search optimal merging ways from three key dimensions: (1) merging approaches, (2) fine-tuning applicability, and (3) merging locations. Specifically, we consider merging patch embeddings at different layers within LVLMs and find that late merging, *i.e.,* merging at the very last layer, preserves most information and achieves minimal performance drop. Additionally, empirical study demonstrates that similarity-oriented merging (clustering) slightly outperforms spatial-oriented merging and resource-efficient fine-tuning (60 A100-GPU hours) further narrows down the performance drop between patch embeddings with/without merging. Combining these findings, we propose a simple but effective baseline about patch-level embeddings reduction in VDR. Extensive experiments on three VDR benchmarks [37, 38, 175] reveal that this baseline achieves comparable performance with ColPali/ColQwen2 models but reduces the patch storage by orders of magnitude. Notably, it retain 98.7% NDCG@5 scores with only 11% of the original memory footprints and 97.5% at 2%, respectively.

7.2 Token Pruning: An Ineffective Strategy

Given patch embeddings E_p for each document page, a natural approach is to retain N_p' embeddings and prune the remaining $(N_p - N_p')$. In this section, we explore four pruning strategies and observe that their performance collapses when reducing embeddings by orders of magnitude. More embarrassingly, the simplest random pruning outperforms most other carefully-designed strategies. Further analysis reveals that ColPali's embeddings are likely in groups, while their relevance to different queries is highly unpredictable. These findings highlight the limitations of pruning strategies and underscore the feasibility and necessity of merging strategies under VDR settings.

7.2.1 Four Pruning Strategies

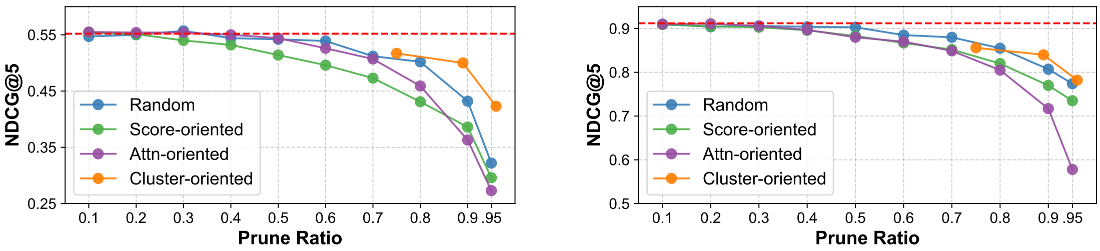
We evaluate four pruning strategies as follows:

Random: For each E_p, we randomly drop $(N_p - N'_p)$ embeddings.

Score-oriented: Recall that ColPali/ColQwen2 measures the relevance between the queries and the pages by maximum-similarity approach, *i.e.,* considering the most similar patch embeddings $e_p^i \in E_p$ with $e_q^j \in E_q$ for each token in q. Accordingly, we denote the *response potential* of each patch $p_i \in p$ on query q as its maximum similarities with any token $q_i \in q$, *i.e.,* $r_p^i(q) = \max_j {e_p^i}^T e_q^j$. However, the key bottleneck for token-reduction in VDR is exactly that the query q, and the associated $r_p^i(q)$, is unknown when we prune E_p at the offline stage. To ensure the performance preservation on any potential q, we prompt LVLMs to generate a set of possible queries Q_p given each document page. Then we approximate the response potential on any queries as the maximum values on this sampled set Q_q: $r_i^p = \max_{q \in Q_p} r_p^i(q)$. We view patches with low r_i^p values as unimportant for any queries and prune them at priority.

Attention-oriented: Recall that the common pruning strategies in LVLM's generation [178, 185] measure the token's importance by their received attentions from other tokens in Transformer layers. We employ this strategy and rank the patch embeddings in E_p by the received attentions (of corresponding tokens in last LVLM layer) from the `[EOS]` token. We prune embeddings with less attentions at priority.

Cluster-oriented: We compute the cosine similarities among the patch embeddings and group them into clusters. For each cluster, We randomly select **ONE** patch in it and prune other patches.



(A) DocVQA (B) InfoVQA

FIGURE 7.2: Retrieval performance v.s. pruning ratio across four different pruning strategies on two datasets.

7.2.2 Results

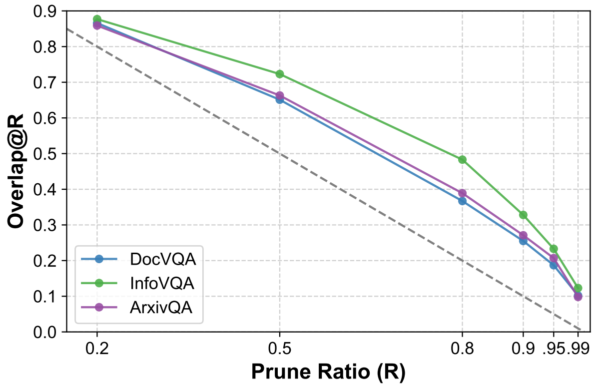
We evaluate the pruning strategies above on two representative datasets, DocVQA [24] and InfoVQA [155], from the ViDoRE [38] benchmark. The embeddings E_p are generated using the official ColQwen2 checkpoints [3] and pruned with varying pruning ratios $(1 - N'_p/N_p)$. The results are shown in Figure 7.2.

All evaluated techniques perform far from satisfactory: Experiments demonstrate that all strategies maintain their NDCG@5 scores when the pruning ratio is below 0.2, and present slight drop ($< 2\%$ absolute score) for ratios below 0.5. However, more aggressive pruning ratios result in significant performance drop. The best-performing strategy retains only 90.9% / 92.3% of its original score at 0.9 pruning ratio and 76.9% / 85.9% at 0.95 ratio, which is far from satisfactory. Overall, none of the four pruning strategies achieves effective token reduction by orders of magnitude.

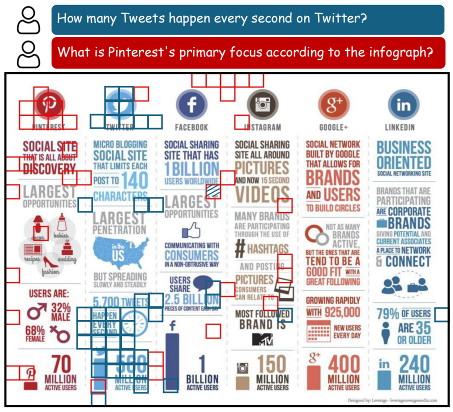
Cluster-oriented pruning performs best across evaluated techniques: It consistently performs best among evaluated strategies under the same pruning ratio, and remains the most resilient as pruning becomes aggressive (ratio 0.7–0.9).

Random dropping is a strong baseline: We surprisingly observe that the simplest random pruning outperforms score-oriented and attention-oriented strategies, especially when the pruning ratio is above 0.5. At 0.95 pruning ratio, it surpasses the score-oriented strategy by 3.9% and the attention-oriented strategy by 19.6% in absolute score on InfoVQA dataset.

[3] `https://huggingface.co/vidore/colqwen2-v1.0`



(A) The activated patches overlap of two queries under different pruning ratios.



(B) A representative case. The activated patches given different queries are colored in red and blue, respectively.

FIGURE 7.3: The triggered patches of the identical page vary with the queries.

7.2.3 Analysis

We investigate the mechanism behind the above observations and attribute it to two possible reasons. Combined with the observations and the analysis, we speculate that token pruning strategies may be inappropriate under VDR setting.

The triggered patches of the identical page vary with the queries. For a document page p, an ideal property in the VDR setting is that the distribution $r_p(q) \in R^{N_p}$ remains consistent across different queries $q \in Q$ (*i.e.*, small $E_q[\mathrm{KL}(r_p(q)||E_q(r_p))]$ value). In other words, we expect significant overlap in the patches activated (having high r_p^i values) by different queries. Being the foundation of pruning strategy, this consistency allows us to accurately predict and retain informative patches with the help of sampled/simulated queries during the offline stage. To quantitatively evaluate the consistency, we use the synthesized queries Q_q given each page p in Section 7.2.1 to compute $r_p(q)$. Then we define the patches in p activated by q as those with top-$K\%$ highest $r_p^i(q)$ values, and pairwise compute the overlap of activated patches by two different queries. We show the overlap at different prune ratios (1-$K\%$) in Figure 7.3(a). It reveals that the shared activated patches of two queries are only marginally higher than what would occur by random chance (in dashed diagonal). A case shown in Figure 7.3(b) further support this result. Given two different queries, the activated patches on the same page are almost exclusive. Only one patch (out of 736; hatched) responds to both queries.

The patch tokens might be grouped. Above experiments demonstrate the (relatively) superiority of clustering-oriented pruning, in which each similarity-clustered group has one and only one patch remaining. And the later experiments in Section 7.3 further show that the cluster-based merged embeddings retain almost all retrieval accuracy. Based on these empirical results, we speculate that the patch tokens might be well-grouped, and leave its further explorations in the future.

We believe that these two reasons inherently limit the effectiveness of pruning strategy under VDR setting where the page embeddings should be pruned offline without access to the queries. Since the activated patches are unpredictable but their representations are grouped, key patches regarding some query are prone to be dropped **by group** according to some specific criterion (like attention- or score-oriented). In such case, they even perform worse than random drop because a group of patches are unlikely to be completely dropped **by random**. If the valuable tokens vary across unpredictable queries, the prudent choice is to conservatively retain all of them (*i.e.,* merging) rather than discard some (*i.e.,* pruning). Therefore, we speculate that pruning strategy is not appropriate for retrieval and turn to exploring token merging strategies.

7.3 Token Merging: The Choices

We turn into another token-reduction strategy, *merging*, towards an efficient document visual retriever. Unlike *pruning* which directly drops some tokens, *merging* consolidates the multiple embeddings into one. This approach is particularly suitable for VDR, where the importance of each embedding is highly undetermined (if not conditioned on specific query). We systematically evaluate the *merging* astrategy through three key aspects towards the recipe for the optimal *merging* strategy as detailed below.

7.3.1 Merging Approach

We follow Clavié et al. [188] and consider four merging approaches as illustrated in Figure 7.4(a) [4].

[4]We do not illustrate the random baseline for the convenience of visualization.

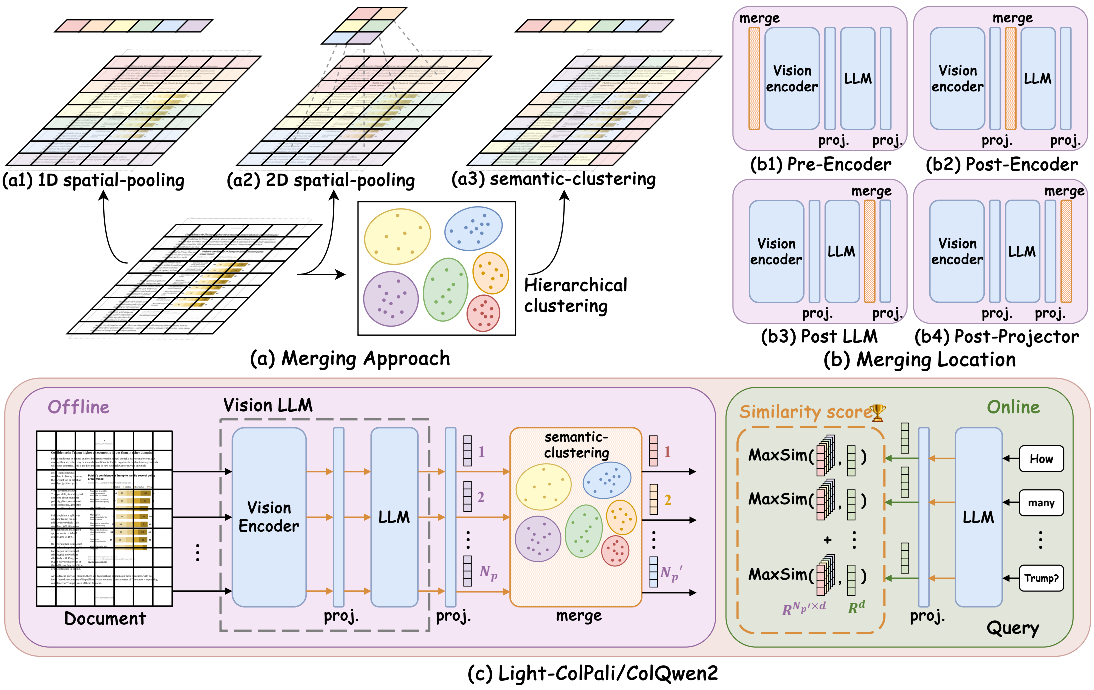


FIGURE 7.4: **(a)**: Three merging approaches. The patches with the same colors are merged into the same embedding. **(b)**: Three merging locations. Blue blocks represent the original modules in ColPali/ColQwen2. Orange blocks represent the added merging modules. **(c)**: The architecture diagram of Light-Colpali/ColQwen2.

Random. The patch embeddings from LVLMs are shaped as $R^{N_p \times d}$. To reduce the embedding numbers from N_p to N_p', we randomly merge every N_p/N_p' embeddings into ones by averagely pooling their representations.

1D Spatial-pooling. In LVLM, images are divided into patches and flattened. A naive (but actually competitive) method is to averagely pool every N_p/N_p' embeddings in sequential order.

2D Spatial-pooling. This approach takes into account the spatial structure and semantics of visualized documents. Building on the intuition that adjacent patches often share semantic relationships, 2D-pooling averagely pools embeddings based on their spatial proximity.

Semantic-clustering. This approach focuses on representation (rather than spatial) proximity. By computing the cosine similarities among the N_p embeddings from ColPali/ColQwen2, we group them into N_p' clusters. Each cluster is then represented by the average of the embeddings within it. In this way, the N_p patch embeddings are merged into N_p' cluster embeddings.

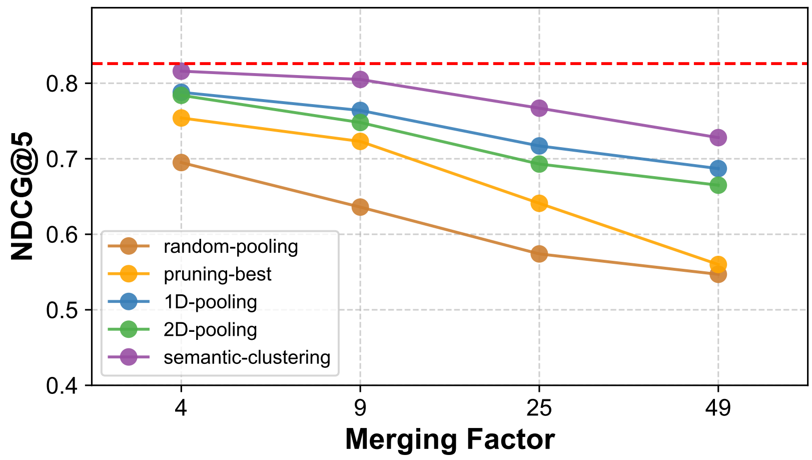


FIGURE 7.5: Performance v.s. merging factor across different approaches. We also show best evaluated pruning (*i.e.*, clustering-oriented pruning) strategy for comparison.

We evaluate the four merging strategies on six datasets from ViDoRE [38] benchmark. We report their average NDCG@5 scores under varying merging factors N_p/N_p' in Figure 7.5. The merging approaches outperform pruning strategies overall, with the clustering approach showing particularly strong results. It maintains 97.5% and 92.6% relative performance at merging factor 9 and 25, respectively. Such results highlight its effectiveness in maintaining retrieval accuracy even under token reduction by orders of magnitude.

7.3.2 Fine-tuning Applicability

Above approaches are training-free and serve as plug-and-play modules for the output patch embeddings. While they achieve promising merging ratios without significant performance degradation, we further investigate whether fine-tuning can enhance the performance maintenance. To this end, we compute the relevance score $s(q, p)$ using the merged document embeddings $E_p' \in R^{N_p' \times d}$ during **BOTH** the training and the inference stage. Results shown in Figure 7.6 show that fine-tuning retrievers with merged embeddings enhances their perceiving on *blurred* representations and reduces their performance gaps with the original retrievers. This benefit is especially significant at large merging factors. Specifically, at merging factors of 25 and 49 (retaining only 4.6% and 2.8% memory cost), fine-tuning recovers 61% and 67% of the performance drop (3.6% and 8.4% absolutely score

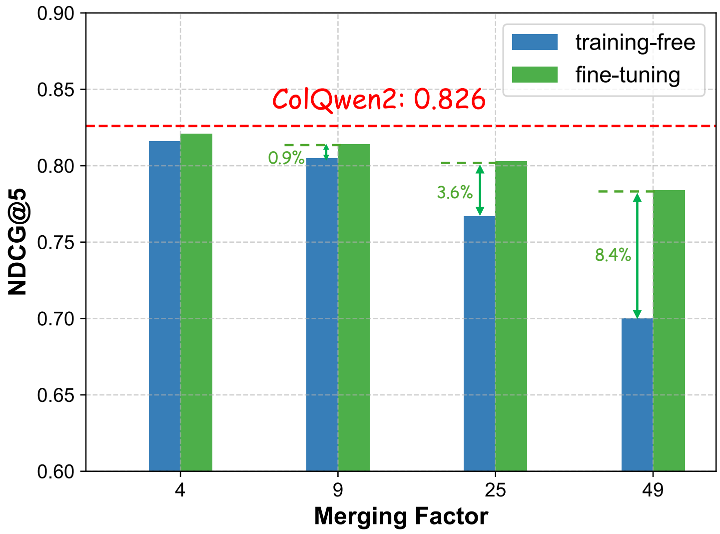


FIGURE 7.6: Training-free v.s. fine-tuning retriever with the same merging approach. The performance of original ColQwen2 is highlighted in red dash.

gains) caused by training-free. Above findings validate the necessity and effectiveness of fine-tuning in maintaining retrieval performance under aggressive token reduction strategies.

7.3.3 Merging Location

We further explore the optimal location of merging operations within ColPali/-ColQwen2. While prior work for efficiency generation [178, 182, 185] typically merges tokens at the early layers of LVLMs to reduce FLOPs and response latency, our focus in VDR setting is primarily on the memory footprint of the offline-stored embeddings. This allows us to consider merging operations at later stages, even if FLOPs and latency remain unchanged or increase slightly. Therefore, we explore inserting merging modules at various locations within ColPali/ColQwen2's architecture. As illustrated in Figure 7.4(b), the four options are: (1) Pre-Encoder, (2) Post-Encoder, (3) Post-LLM and (4) Post-projector.

We compare the performance of different merging locations at merging factor 9 in Table 7.1. We observe that (1) performance significantly improves when the merging operation occurs after LLM module. It demonstrates that token reduction should be performed as late as possible when FLOPs and latency are not the concern, as feeding more visual tokens to the LLM allows for finer-grained

Table 7.1: Retrieval performance of different merging locations.

	Pre-Encoder	**Post-Encoder**	**Post-LLM**	**Post-Projector**
Info	70.2	79.5	89.7	90.4
Doc	29.8	41.7	55.2	56.1
Arxiv	80.0	81.9	87.6	86.7
TabF	74.1	80.8	88.6	88.8
TAT	50.5	54.1	79.5	79.1
Shift	49.7	54.4	85.7	87.3
Avg.	59.1	65.4	81.0	81.4

perception and more accurate information integration. (2) merging after the final projector yields slightly better performance (0.4% absolute score) than before it. Since the projector is designed for dimension reduction (*e.g.,* from 1536 to 128 for ColQwen2), we hypothesize that clustering algorithms are more effective in low-dimension spaces and thus enable more targeted feature aggregation.

7.4 Light-ColQwen2: Effective Storage Reduction on Patch-level Embeddings

We conduct extensive experiments to identify the optimal merging strategy in Section 7.3. The key findings are as follows: (1) *Merging Approach*: Merging upon representation similarity (semantic clustering) outperforms spatial proximity (1D- / 2D-spatial pooling). (2) *Merging Location*: Merging at the last stage of retrievers fully leverages the powerful perception capabilities of LVLMs and thus achieves minimal performance drop. (3) *Fine-tuning Applicability*: Incorporating the merging module during training stage significantly reduces the gap compared to the original retrievers, particularly at high reduction ratios.

Based on these insights, we propose a simple yet effective token-reduction approach for ColPali/ColQwen2, named Light-ColPali/ColQwen2. As illustrated in Figure 7.4(c), it is a token merging strategy which integrates semantic clustering at the latest stage of the pipeline, combined with fine-tuning, to achieve efficient and accurate visualized document retrieval. The simplicity and effectiveness of

TABLE 7.2: The NDCG@5 scores of different visualized document retrievers on three benchmarks. We report their average scores at the most right column, with their relative performance compared with the original ColPali/ColQwen2. We also report their relative memory costs (**# Mem**) compared with DSE-Pali/Qwen2.

	Merging Factor	**# Mem**	**ViDoRE**						**VisRAG**		**MM-LB**	**Average**
			Info	**Doc**	**Arxiv**	**TabF**	**TAT**	**Shift**	**Slide**	**Chart**		
Base model: Qwen2-VL-2B (original patch number: 768)												
DSE-Qwen2	-	1.0	84.7	50.0	84.6	89.2	67.1	78.5	86.8	57.6	68.0	$74.1_{91.0\%}$
ColQwen2	-	64.4	91.5	55.4	88.0	90.5	81.1	88.5	93.4	65.8	78.6	$81.4_{100.0\%}$
ColQwen2$_{+\text{Pruning}}$	9	7.6	85.6	48.3	84.0	88.3	68.6	72.5	89.3	60.3	69.0	$74.0_{90.9\%}$
	49	1.8	74.7	36.3	77/1	80.5	46.7	55.9	77.3	52.8	62.3	$62.6_{76.9\%}$
Light-ColQwen2	4	16.4	89.5	56.6	88.6	90.2	80.5	87.1	92.9	62.9	77.0	$80.6_{99.0\%}$
	9	7.6	90.4	56.1	86.7	88.8	79.1	87.3	92.2	62.0	76.2	$79.9_{98.2\%}$
	25	3.0	88.9	54.6	86.4	89.3	78.7	84.4	91.0	60.4	71.9	$78.4_{96.3\%}$
	49	1.8	86.9	52.6	86.5	86.8	73.5	84.5	89.7	59.6	72.8	$77.0_{94.6\%}$
Base model: PaliGemma-3B (original patch number: 1024)												
DSE-Pali	-	1.0	80.1	46.0	82.0	84.1	61.1	70.2	84.8	54.7	67.0	$70.0_{91.5\%}$
ColPali	-	36.7	84.4	54.8	85.1	85.3	72.3	75.5	92.2	62.0	77.1	$76.5_{100.0\%}$
ColPali$_{+\text{Pruning}}$	9	4.2	81.5	50.5	82.0	84.4	61.1	67.0	90.2	59.0	69.1	$71.6_{93.6\%}$
	49	0.9	72.5	35.8	70.3	72.6	40.3	44.1	79.1	50.3	61.9	$58.6_{76.6\%}$
Light-ColPali	4	9.3	82.8	53.4	84.1	86.5	72.8	72.5	91.7	60.6	73.3	$75.3_{98.4\%}$
	9	4.2	82.1	54.8	83.5	84.5	70.9	72.8	91.2	61.0	72.6	$74.8_{97.8\%}$
	25	1.6	81.2	50.5	82.6	82.7	67.2	70.7	90.8	57.3	71.9	$72.8_{95.2\%}$
	49	0.9	79.9	49.6	82.7	81.9	67.4	69.0	88.9	57.5	68.8	$71.6_{93.6\%}$

Light-ColPali/ColQwen2 make it a practical solution for balancing performance and efficiency in visual document retrieval tasks.

Baseline We evaluate Light-ColPali/ColQwen2 against three primary baselines. (1) The original ColPali/ColQwen2 [38] which encodes each patch in the page as one embedding. (2) DSE-Pali/-Qwen2 [39] which encodes each page into one embedding. (3) The most effective pruning strategy, random pruning, as introduced in Section 7.2.1. We compare them in terms of both retrieval performance and memory cost.

Experiment Setup We conduct experiments on nine datasets from three benchmarks: ViDoRE [38], VisRAG [175] and MMLongBench-Doc [37]. We follow previous work to use NDCG@5 as the evaluation metric on performance and relative memory cost (compared with DSE) as the metric on efficiency.

Result Based on Qwen2-VL-2B [149] and PaliGemma-3B [174], we show results of different visualized document retrievers on Figure 7.1 and Table 7.2. We observe that (1) ColPali/ColQwen2 achieves superior performance but comes at the cost of significantly larger memory footprint than DSE. Specifically, ColPali/Qwen2 outperforms DSE by 6.7% absolute scores on Qwen2-VL-2B and 6.5% absolute scores on PaliGemma-3B. However, they also takes up more than 64.4 and 36.7 times of memory cost and cause big burdens on both offline indexing and online retrieval. Above results highlight the necessity for a performance-cost balance. (2) Light-ColPali/ColQwen2 achieves a significant reduction in memory footprint while largely preserving performance. For Light-ColQwen2, it maintains 99.0% of NDCG@5 scores (80.6 out of 81.4) at a merging factor of 4 and 98.2% of NDCG@5 scores at a merging factor of 9. Even at an extremely large merging ratio, where its memory cost is comparable to DSE (1.8x), Light-ColQwen2 retains 94.5% relative performance and outperforms DSE by 2.9% in absolute score gains. Similarly, Light-ColPali maintains 98.4% and 97.8% of NDCG@5 scores at merging factors of 4 and 9, respectively. Furthermore, at an extreme reduction ratio of 49 (even lower memory cost than DSE), Light-ColPali retains 93.6% relative performance and surpasses DSE by 1.6% in absolute score gains. These results demonstrate that Light-ColPali/ColQwen2 effectively balances memory efficiency and retrieval performance. (3) Light-ColPali/ColQwen2 presents various performance across different datasets. For InfoVQA, ArxivQA, TabFQuAD and SlideVQA where documents typically have lower information densities (*e.g.,* posters, diagrams), the performance retention is notably higher. In contrast, for datasets like DocVQA, TAT-DQA, and ChartQA where documents are more text-rich and incorporates more information, the performance drop is slightly more obvious. We speculate that the optimal merging factor for each document page highly correlates with its information density. However, how to adaptively adjust the merging factor, both during training and inference stage, remains an open challenge.

Time Cost (Offline Stage) The clustering operation in Light-ColPali/ColQwen2 incurs a modest additional time cost during both model training and embedding generation in the offline stage. As shown in Table 7.3, it adds 3-3.5 hours to the training time and 0.9 minute to the document embedding generation time. We consider this slight increase in offline latency acceptable given the substantial reduction in memory footprint and the resulting acceleration during the online retrieval stage.

TABLE 7.3: Time cost of ColQwen2 v.s. Light-ColQwen2 during offline stage. **Training**: 5 epochs (2310 steps with batch size 128) on 8 A100 GPUs. **Embed Gen**: 500 page embeddings on single A100 GPU.

Model	**ColQwen2**		**Light-ColQwen2**	
	2B	7B	2B	7B
Training	5.6 h	7.5 h	9.0 h	10.5 h
Embed Gen	1.7 min	2.1 min	2.6 min	3.0 min

7.5 Conclusion

This work conducts an empirical study into developing efficient visualized document retrievers with minimal memory footprints and performance loss. Through comprehensive experiments, we demonstrate the superior performance of token merging strategy for VDR tasks. Upon this finding, we propose a simple yet effective merging strategy named Light-ColPali/ColQwen2. It maintains 98.2% of retrieval performance with only 11.8% of original memory usage, and preserves 94.6% effectiveness at 2.8% memory footprint. We believe this established baseline provides valuable insights for advancing efficient VDR research under multi-modal, large-scale settings.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The excessive documents in the information age make automatic document understanding in urgent needs. Towards practical automatic document understanding, this thesis presents a comprehensive investigation along the four primary challenges: **Few-shot**, **Large-scale**, **Long-context**, and **Multi-modal** mentioned in Chapter 1. After a brief summary of previous work in Chapter 2, we address these challenges in the two sub-tasks of document understanding, *i.e.,* Information Extraction and Document Reading Comprehension.

The first part focuses on Information Extraction (IE). Chapter 3 addresses the **few-shot** challenge in event detection by establishing a unified framework and proposing a simple yet effective *baseline*. This baseline significantly outperforms existing methods. Chapter 4 extends our study to event argument extraction and handles both **few-shot** and **large-scale** challenges through a novel prompt tuning paradigm. The resulting model (PAIE) effectively and efficiently extracts event arguments in the document. Chapter 5 further explores the broader application of LLMs to IE tasks. While empirical studies show that LLMs are not good information extractors, we introduce a *filter-then-rerank* paradigm that leverages the strengths of both SLMs and LLMs. This paradigm achieves state-of-the-art performance in **few-shot** and **large-scale** IE scenarios.

The second part focuses on document reading comprehension. Chapter 6 evaluates LVLMs' capabilities on **multi-modal**, **long-context** documents. To this end, we construct a high-quality benchmark named MMLongBench-Doc. While revealing promising potential, our analysis demonstrates significant challenges in lengthy document understanding. To bypass these limitations, we dive deep into visual document retrieval (VDR) and pursue efficient retrieval for **large-scale** applications in Chapter 7. After identifying the prohibitive computational bottlenecks in existing VDR approaches, we conduct extensive empirical study and summarize a token merging strategy that achieves substantial memory reduction with minimal performance loss. It offers critical insights for advancing VDR research in **multi-modal**, **large-scale** scenarios.

In conclusion, this thesis makes significant contributions to practical automatic document understanding from multiple dimensions. We conduct empirical studies, construct new benchmarks, and develop innovative approaches to meet the real-world applications of both IE and reading comprehension tasks. Though there are lots of remaining open problems (detailed in the next section), we believe that these contributions offer valuable insights and lay a solid foundation for future research towards more robust and scalable document understanding systems.

8.2 Future Work

8.2.1 More Flexible Extraction

All IE tasks discussed in this thesis rely on well-defined schemas. However, we argue that such schemas are not essential and can be relaxed through two approaches: (1) automatic schema induction, which avoids the manual schema construction, and (2) personalized extraction, which allows for ad-hoc schema based on user needs. We detail their ideas as follows.

Automatic Schema Induction. Manual schema construction is costly, time-consuming, and requires expertise. Moreover, schemas designed for one domain often fail when applied to another. Here raises a natural question: Can systems autonomously discover and construct schemas conditioned on the document corpora? Prior work [256–258] has explored this through clustering-based methods,

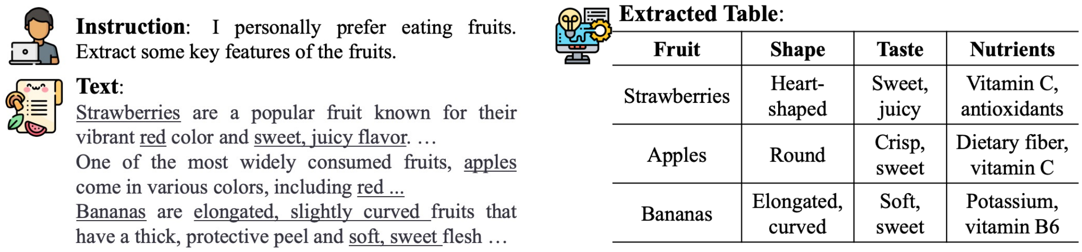


FIGURE 8.1: The paradigm of personalized extraction. Figure curated from [2].

using document corpora and linguistic resources like Abstract Meaning Representation [259], WordNet [260], and FrameNet [261]. Specifically, these methods induce schema structures by clustering candidate knowledge representations. However, the naming of schema in these approaches highly relies on existing lexical knowledge, which might be insufficient, inflexible and outdated. Additionally, clustering-based methods naturally struggle on the discovery of rare but valuable information.

The power of LLMs can address the limitations of previous schema induction algorithms. On one hand, LLMs can generate the schema name in an end-to-end way. On the other hand, the inherent knowledge in LLMs can significantly benefit the detection of rare information. The challenges of LLM-based schema induction is (1) how to control the granularity of the schema taxonomy, (2) how to remove the duplicated names of the candidate names, and (3) how to select the involved document to balance the efficiency and schema quality. We have witnessed related work in automatic tagging [262], and the similar pipeline can be applied to IE domain for the automatic schema induction.

Personalized Extraction. User needs vary across documents and intentions. Pre-defined schemas (either manually- or automatically-constructed) often fail to meet these diverse needs. Moreover, user queries can be ambiguous and require further clarification of intent. Above challenges lead to the emergence of personalized extraction where schemas are generated ad-hoc based on the given documents. For example, On-demand IE [2] (illustrated in Figure 8.1) identifies user intentions (key features of the fruits) and generates one-time schemas (shape, taste, nutrient) tailored to specific needs. Furthermore, some following works further extend this paradigm to specific domains like finance [263].

Despite its potential, personalized extraction faces two challenges. Firstly, evaluating highly flexible schemas is difficult because ground-truth schemas depend on user instructions and are hard to annotate. As a result, the lack of evaluation metrics (and benchmarks) limits its adoption. Secondly, the real user intentions are often hidden behind their explicit instructions. So the performance of personalized extraction systems heavily depends on the reasoning capabilities of LLMs, which remains an open problem (detailed in the next sections) and is critical for enhancing user experiences with personalized extractors.

8.2.2 Reasoning on Documents

In theory, an ideal reading comprehension system should smoothly handle questions at different levels and requiring various abilities, including perception, localization, reasoning, *etc.* Although document reading comprehension has been largely expanded and reshaped by LLMs and LVLMs, current document systems fall short in reasoning capabilities, particularly when reasoning across different pages and modalities. As shown in Section 6.4, even the best-performing model (GPT-4o) demonstrates strong long-context understanding but struggles with many reasoning-heavy questions in the MMLongBench-Doc benchmark.

Recent progress in reasoning models [264, 265] has shown promising results in mathematical and coding tasks. These tasks feature deterministic results and enable reinforcement learning with sparse rewards from simple, rule-based verifiers. However, the learning paradigm of these reasoning models cannot be directly transferred to document-related domains. Unlike math and coding problems, queries in document understanding often require open-ended answers. Additionally, the correctness of reasoning steps (which users are likely read by users) is also critical. This necessitates the use of process reward models (PRMs) to provide fine-grained feedback on model responses.

Developing effective PRMs is challenging for at least two reasons. (1) High-quality, step-wise training samples are time-consuming and expensive to annotate. (2) PRMs are easily hacked and often fail to accurately evaluate the true quality of model responses. Addressing these challenges could significantly enhance the reading comprehension capabilities of LLMs/LVLMs (as well as most other practical tasks without deterministic answers). We believe this area is worthy of further exploration in the future.

8.2.3 AI Agents for Documents

This thesis focuses primarily on document understanding, where systems act as brains and extract knowledge, answer queries, even perform reasoning. Although they have benefitted various document-related domains, we believe that their powers can be further expanded if they can act as the hands of humans, *i.e.,* autonomously executing actions upon their understanding and decisions. We refer to such systems as AI agents.

AI agents for documents have a wide range of real-world applications. For example, they could (1) update bookkeeping sheets with recent transaction records [266], (2) read programming tutorials and fix bugs in code repositories [267], (3) analyze financial reports and execute stock trading operations [6], (4) compare multiple websites (which, in a broader sense, are also documents) and purchase the most suitable flight tickets [268]. We suggest the future work incorporating diverse agents into powerful document system to bridge the gap between understanding and action. Additionally, we call for the future explorations on enhancing the robustness and scalability capabilities of these agents.

8.2.4 Real-world Deployment

While this thesis studies scalability from the algorithm perspective, deploying document understanding systems in production introduces additional problems. We detail them as follows and leave further exploration in the future.

Small-large Model Collaboration. Production pipelines must satisfy strict end-to-end service level objectives (SLOs) under bounded budgets. Building on our results in joint/prompt-based extraction and candidate filtering, a practical path is to adopt a *filter-then-rerank* design: use smaller models to prune easy cases and reserve large models for the hard tail. Future work may include adaptive routing policies learned from live traffic and caching strategies to amortize repeated documents.

Index Growth and Retrieval Efficiency. Large multi-page, multi-modal corpora make vector indexes the dominant serving cost. Our analysis of patch/token-level trade-offs suggests merging or compression at indexing time to reduce memory and I/O while retaining retrieval quality. Future work can be: (i) collection-specific merging factors, (ii) end-to-end time accounting, and (iii) dynamic, low-disruption re-indexing schedules for frequently updated sources.

Bias in LLM-based Reranking. We acknowledge that reranking with LLMs can amplify upstream biases. To mitigate these risks, we recommend the following practices: (1) *optional swapping*: shuffling the option orders can significantly decrease the bias from the MCQ (multi-choice question) format. (ii) *group-wise evaluation*: monitoring reranking metrics across salient cohorts (such as document source and language) and controlling the reranking threshold if necessary.

Privacy Risks with Sensitive Documents. Medical, law and financial documents typically contain confidential data. Responsible deployment should follow *data minimization* (process only necessary pages/spans) and log accesses/model actions for accountability. Additionally, fine-tuning on user content by default should be avoided as much as possible.

List of Author's Publications[1]

Publications

- **Yubo Ma**, Jinsong Li, Yuhang Zang, Xiaobao Wu, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Haodong Duan, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2025. Towards Storage-Efficient Visual Document Retrieval: An Empirical Study on Reducing Patch-Level Embeddings. *In Findings of the Association for Computational Linguistics: ACL 2025*, pages 19568–19580, Vienna, Austria. Association for Computational Linguistics.

- **Yubo Ma**, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. MMLongBench-Doc: Benchmarking Long-context Document Understanding with Visualizations. *38th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, pages 95963-96010, Vancouver, Canada. Curran Associates, Inc.

- **Yubo Ma**, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. 2024. SciAgent: Tool-augmented Language Models for Scientific Reasoning. *In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15701–15736, Miami, Florida, USA. Association for Computational Linguistics.

- **Yubo Ma**, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker

[1] The superscript * indicates joint first authors

for Hard Samples!. *In Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601, Singapore. Association for Computational Linguistics.

- **Yubo Ma**, Zehao Wang, Yixin Cao, and Aixin Sun. 2023. Few-shot Event Detection: An Empirical Study and a Unified View. *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11211–11236, Toronto, Canada. Association for Computational Linguistics.

- **Yubo Ma***, Zehao Wang*, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

- **Yubo Ma***, Zehao Wang*, Mukai Li*, Yixin Cao*, Meiqi Chen*, Xinze Li, Wenqi Sun, Kunquan Deng, Kun Wang, Aixin Sun, and Jing Shao. 2022. MMEKG: Multi-modal Event Knowledge Graph towards Universal Representation across Modalities. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 231–239, Dublin, Ireland. Association for Computational Linguistics.

- Shumin Deng, **Yubo Ma**, Ningyu Zhang, Yixin Cao, and Bryan Hooi. 2024. Information Extraction in Low-Resource Scenarios: Survey and Perspective. *IEEE International Conference on Knowledge Graph (ICKG)*, pages 33-49, Abu Dhabi, United Arab Emirates.

- Zehao Wang, Minye Wu, Yixin Cao, **Yubo Ma**, Meiqi Chen, and Tinne Tuytelaars. 2024. Navigating the Nuances: A Fine-grained Evaluation of Vision-Language Navigation. *In Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4681–4704, Miami, Florida, USA. Association for Computational Linguistics.

- Meiqi Chen, **Yubo Ma**, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2024. Improving Large Language Models in Event Relation Logical Prediction. *In Proceedings of the 62nd Annual Meeting of the Association* *for Computational Linguistics (Volume 1: Long Papers)*, pages 9451–9478, Bangkok, Thailand. Association for Computational Linguistics.

- Xinze Li, Yixin Cao, Liangming Pan, **Yubo Ma**, and Aixin Sun. 2024. Towards Verifiable Generation: A Benchmark for Knowledge-aware Language Model Attribution. *In Findings of the Association for Computational Linguistics: ACL 2024*, pages 493–516, Bangkok, Thailand. Association for Computational Linguistics.

Bibliography

[1] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *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. xx, 70

[2] 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/. xxi, 17, 117

[3] Jaromir Savelka and Kevin Ashley. Discovering explanatory sentences in legal case decisions using pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4273–4283, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.361. URL https://aclanthology.org/2021.findings-emnlp.361. 2

[4] Ruihao Shui, Yixin Cao, Xiang Wang, and Tat-Seng Chua. A comprehensive evaluation of large language models on legal judgment prediction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7337–7348, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.490. URL https://aclanthology.org/2023.findings-emnlp.490/. 2

[5] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic, November 2021. Association for

Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300. 2, 18

[6] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 4314–4325, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671801. URL https://doi.org/10.1145/3637528.3671801. 2, 119

[7] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications: A literature review. *Journal of Biomedical Informatics*, 77:34–49, 2018. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2017.11.011. URL https://www.sciencedirect.com/science/article/pii/S1532046417302563. 2

[8] Lutz Bornmann and Rüdiger Mutz. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, 66, 2014. URL https://api.semanticscholar.org/CorpusID:7826703. 2

[9] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1360. URL https://aclanthology.org/D18-1360. 2

[10] Shumin Deng, Yubo Ma, Ningyu Zhang, Yixin Cao, and Bryan Hooi. Information extraction in low-resource scenarios: Survey and perspective. In *2024 IEEE International Conference on Knowledge Graph (ICKG)*, pages 33–49, 2024. doi: 10.1109/ICKG63256.2024.00013. 3, 11

[11] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition : Extended abstract. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3817–3818, 2023. doi: 10.1109/ICDE55515.2023.00335. 3, 11

[12] Elisa Bassignana and Barbara Plank. What do you mean by relation extraction? a survey on datasets and study on scientific relation classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 67–83, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-srw.7. URL https://aclanthology.org/2022.acl-srw.7. 3

[13] Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, and Philip S. Yu. A survey on deep learning event extraction: Approaches and applications. *IEEE Trans. Neural Networks Learn. Syst.*, pages 1–21, 2022. doi: 10.1109/TNNLS.2022.3213168. URL `https://ieeexplore.ieee.org/abstract/document/9927311`. 3, 12

[14] Xin 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 *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 601–610, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623623. URL `https://doi.org/10.1145/2623330.2623623`. 3

[15] Manling Li, Alireza Zareian, Ying Lin, Xiaoman Pan, Spencer Whitehead, Brian Chen, Bo Wu, Heng Ji, Shih-Fu Chang, Clare Voss, Daniel Napierski, and Marjorie Freedman. GAIA: A fine-grained multimedia knowledge extraction system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 77–86, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.11. URL `https://aclanthology.org/2020.acl-demos.11`.

[16] Haoyang Wen, Yanru Qu, Heng Ji, Qiang Ning, Jiawei Han, Avi Sil, Hanghang Tong, and Dan Roth. Event time extraction and propagation via graph attention networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 62–73, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.6. URL `https://aclanthology.org/2021.naacl-main.6`.

[17] Xinya Du, Zixuan Zhang, Sha Li, Pengfei Yu, Hongwei Wang, Tuan Lai, Xudong Lin, Ziqi Wang, Iris Liu, Ben Zhou, Haoyang Wen, Manling Li, Darryl Hannan, Jie Lei, Hyounghun Kim, Rotem Dror, Haoyu Wang, Michael Regan, Qi Zeng, Qing Lyu, Charles Yu, Carl Edwards, Xiaomeng Jin, Yizhu Jiao, Ghazaleh Kazeminejad, Zhenhailong Wang, Chris Callison-Burch, Mohit Bansal, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, Martha Palmer, and Heng Ji. RESIN-11: Schema-guided event prediction for 11 newsworthy scenarios. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 54–63, Hybrid: Seattle, Washington + Online, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-demo.7. URL `https://aclanthology.org/2022.naacl-demo.7`. 3

[18] Jordan Boyd-Graber and Benjamin Börschinger. What question answering can learn from trivia nerds. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7422–7435, Online, July

2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.662. URL https://aclanthology.org/2020.acl-main.662. 3

[19] Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. DeFormer: Decomposing pre-trained transformers for faster question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4497, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.411. URL https://aclanthology.org/2020.acl-main.411. 3

[20] Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2138–2149, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.184. URL https://aclanthology.org/2021.findings-emnlp.184. 3

[21] Qintong Li, Piji Li, Wei Bi, Zhaochun Ren, Yuxuan Lai, and Lingpeng Kong. Event transition planning for open-ended text generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3412–3426, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.269. URL https://aclanthology.org/2022.findings-acl.269. 3

[22] Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. A survey on machine reading comprehension systems, 2020. URL https://arxiv.org/abs/2001.01582. 3

[23] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264. 4, 18

[24] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2199–2208, 2021. doi: 10.1109/WACV48630.2021.00225. 4, 20, 84, 105

[25] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259. 4, 18, 20

[26] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL `https://aclanthology.org/2022.acl-long.229`. 4

[27] Fei Wu, Raphael Hoffmann, and Daniel S. Weld. Information extraction from wikipedia: moving down the long tail. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 731–739, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1401978. URL `https://doi.org/10.1145/1401890.1401978`. 4

[28] 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, 2024. URL `https://arxiv.org/abs/2303.18223`. 5

[29] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 11 2024. ISSN 2095-5138. doi: 10.1093/nsr/nwae403. URL `https://doi.org/10.1093/nsr/nwae403`. 5, 7

[30] 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/`. 5

[31] Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Xiangkun Hu, Zheng Zhang, Qian Wang, and Yue Zhang. Novelqa: Benchmarking question answering on documents exceeding 200k tokens, 2024. URL `https://arxiv.org/abs/2403.12766`. 5

[32] Yubo Ma, Zehao Wang, Yixin Cao, and Aixin Sun. Few-shot event detection: An empirical study and a unified view. 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 11211–11236, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.628. URL `https://aclanthology.org/2023.acl-long.628/`. 6, 27

[33] Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 6759–6774, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.466. URL https://aclanthology.org/2022.acl-long.466. 6, 45, 66

[34] 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 of Computer Science*, 18(6):186357, 2024. 6

[35] Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. A survey of generative information extraction. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4840–4870, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.324/. 6

[36] Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.710. URL https://aclanthology.org/2023.findings-emnlp.710/. 7, 63

[37] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. MMLONGBENCH-DOC: Benchmarking long-context document understanding with visualizations. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=loJM1acwzf. 7, 84, 103, 112

[38] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, CELINE HUDELOT, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ogjBpZ8uSi. 7, 22, 24, 102, 103, 105, 109, 112

[39] Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. Unifying multimodal retrieval via document screenshot embedding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6492–6505, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.373. URL https://aclanthology.org/2024.emnlp-main.373/. 7, 21, 112

[40] Yubo Ma, Jinsong Li, Yuhang Zang, Xiaobao Wu, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Haodong Duan, Jiaqi Wang, Yixin Cao, and Aixin Sun. Towards storage-efficient visual document retrieval: An empirical study on reducing patch-level embeddings. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19568–19580, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1003. URL `https://aclanthology.org/2025.findings-acl.1003/`. 8, 101

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`. 11

[42] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL `https://aclanthology.org/2020.acl-main.703`. 11, 31, 49

[43] 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`. 11, 31

[44] OpenAI. Gpt-4 technical report, 2023. 11, 63, 84, 93

[45] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large

language models trained on code. *CoRR*, abs/2107.03374, 2021. URL `https://arxiv.org/abs/2107.03374`. 11, 67

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 11, 17, 63, 67

[47] Ningyu Zhang, Shumin Deng, Zhen Bi, Haiyang Yu, Jiacheng Yang, Mosha Chen, Fei Huang, Wei Zhang, and Huajun Chen. OpenUE: An open toolkit of universal extraction from text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 1–8, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.1. URL `https://aclanthology.org/2020.emnlp-demos.1`. 11

[48] Ningyu Zhang, Xin Xu, Liankuan Tao, Haiyang Yu, Hongbin Ye, Shuofei Qiao, Xin Xie, Xiang Chen, Zhoubo Li, and Lei Li. DeepKE: A deep learning based knowledge extraction toolkit for knowledge base population. In *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 98–108, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-demos.10`. 11

[49] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: a large margin approach. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 896–903, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102464. URL `https://doi.org/10.1145/1102351.1102464`. 11

[50] Hailin Wang, Ke Qin, Rufai Yusuf Zakari, Guoming Lu, and Jin Yin. Deep neural network-based relation extraction: an overview. *Neural Comput. Appl.*, 34(6):4781–4801, 2022. doi: 10.1007/s00521-021-06667-3. URL `https://doi.org/10.1007/s00521-021-06667-3`. 12

[51] Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 28–40, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.3. URL `https://aclanthology.org/2021.findings-acl.3`. 13, 15, 28, 31, 34, 39

[52] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4080–4090, Red Hook, NY, USA, 2017. ISBN 9781510860964. 14, 15, 31, 34

[53] Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. Few-shot classification in named entity recognition task. New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359337. 14

[54] Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.128. URL `https://aclanthology.org/2020.acl-main.128`. 14, 31, 34, 39

[55] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. 14

[56] Yi Yang and Arzoo Katiyar. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.516. URL `https://aclanthology.org/2020.emnlp-main.516`. 14, 29

[57] Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.439. URL `https://aclanthology.org/2022.acl-long.439`. 14, 31, 34, 41

[58] Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.155. URL `https://aclanthology.org/2022.findings-acl.155`. 14, 29, 31, 34, 66

[59] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034. 14

[60] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1129. URL https://aclanthology.org/P19-1129. 14

[61] Jiashu Xu, Mingyu Derek Ma, and Muhao Chen. Can NLI provide proper indirect supervision for low-resource biomedical relation extraction? 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 2450–2467, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.138. URL https://aclanthology.org/2023.acl-long.138/. 14

[62] Keming Lu, I-Hung Hsu, Wenxuan Zhou, Mingyu Derek Ma, and Muhao Chen. Summarization as indirect supervision for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6575–6594, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.490. 14, 75

[63] Yang Liu, Jinpeng Hu, Xiang Wan, and Tsung-Hui Chang. A simple yet effective relation information guided approach for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 757–763, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.62. URL https://aclanthology.org/2022.findings-acl.62. 14

[64] 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, 2022. doi: 10.1145/3485447.3511998. URL https://doi.org/10.1145/3485447.3511998. 66

[65] Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. Few-shot relation extraction via Bayesian meta-learning on relation graphs. In

Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7867–7876. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/qu20a.html.

[66] Wen Wen, Yongbin Liu, Chunping Ouyang, Qiang Lin, and Tonglee Chung. Enhanced prototypical network for few-shot relation extraction. *Information Processing Management*, 58(4):102596, 2021. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2021.102596. URL https://www.sciencedirect.com/science/article/pii/S0306457321000959.

[67] Chih-Yao Chen and Cheng-Te Li. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.272. URL https://aclanthology.org/2021.naacl-main.272. 14

[68] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In *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. 14, 15, 30, 55, 56

[69] Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. Event extraction as machine reading comprehension. In *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. 15, 28, 30

[70] Rui Feng, Jie Yuan, and Chao Zhang. Probing and fine-tuning reading comprehension models for few-shot event extraction. *CoRR*, abs/2010.11325, 2020. URL https://arxiv.org/abs/2010.11325. 14, 28

[71] Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.42. URL https://aclanthology.org/2021.acl-short.42. 14, 28, 30

[72] Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. Structured prediction as translation between augmented

natural languages. 2021. doi: 10.48550/ARXIV.2101.05779. URL https://arxiv.org/abs/2101.05779. 14, 15

[73] 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 *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. 15, 28

[74] 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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.395. URL https://aclanthology.org/2022.acl-long.395. 28, 30, 66

[75] 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 *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. 14, 28, 30

[76] Sha Li, Liyuan Liu, Yiqing Xie, Heng Ji, and Jiawei Han. Piled: An identify-and-localize framework for few-shot event detection, 2022. URL https://arxiv.org/abs/2202.07615. 14, 28

[77] Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. Exploiting the matching information in the support set for few shot event classification. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020.*, 2020. 15

[78] Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nuse-1.5. URL https://aclanthology.org/2020.nuse-1.5.

[79] Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page

151–159, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371796. URL `https://doi.org/10.1145/3336191.3371796`. 28

[80] Shumin Deng, Ningyu Zhang, Luoqiu Li, Chen Hui, Tou Huaixiao, Mosha Chen, Fei Huang, and Huajun Chen. OntoED: Low-resource event detection with ontology embedding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2828–2839, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.220. URL `https://aclanthology.org/2021.acl-long.220`. 15, 28

[81] Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. Learning prototype representations across few-shot tasks for event detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5270–5277, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.427. URL `https://aclanthology.org/2021.emnlp-main.427`. 15, 28

[82] Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1201. URL `https://aclanthology.org/P18-1201`. 15, 28

[83] Shirong Shen, Tongtong Wu, Guilin Qi, Yuan-Fang Li, Gholamreza Haffari, and Sheng Bi. Adaptive knowledge-enhanced Bayesian meta-learning for few-shot event detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2417–2429, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.214. URL `https://aclanthology.org/2021.findings-acl.214`. 15

[84] Kailin Zhao, Xiaolong Jin, Long Bai, Jiafeng Guo, and Xueqi Cheng. Knowledge-enhanced self-supervised prototypical network for few-shot event detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6266–6275, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.findings-emnlp.467`. 15, 28

[85] Ruihan Zhang, Wei Wei, Xian-Ling Mao, Rui Fang, and Dangyang Chen. HCL-TAT: A hybrid contrastive learning method for few-shot event detection with task-adaptive threshold. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1808–1819, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.findings-emnlp.130`. 15, 28

[86] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006. 15, 39

[87] Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4672–4682, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.360. URL `https://aclanthology.org/2021.acl-long.360`. 15, 55, 56

[88] Jian Liu, Yufeng Chen, and Jinan Xu. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *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`. 15, 55, 56

[89] Sha Li, Heng Ji, and Jiawei Han. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.69. URL `https://aclanthology.org/2021.naacl-main.69`. 15, 49, 54, 55, 56

[90] Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. Exploring the feasibility of chatgpt for event extraction. *CoRR*, abs/2303.03836, 2023. URL `https://doi.org/10.48550/arXiv.2303.03836`. 15, 16

[91] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. Zero-shot information extraction via chatting with chatgpt. *CoRR*, abs/2302.10205, 2023. URL `https://doi.org/10.48550/arXiv.2302.10205`. 15, 16

[92] Zizheng Lin, Hongming Zhang, and Yangqiu Song. Global constraints with prompting for zero-shot event argument classification. In *EACL (Findings)*, pages 2482–2493. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.findings-eacl.191`. 15, 16

[93] Guozheng Li, Peng Wang, and Wenjun Ke. Revisiting large language models as zero-shot relation extractors. In *EMNLP (Findings)*, pages 6877–6892. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.findings-emnlp.459`. 15, 16

[94] Dhananjay Ashok and Zachary C. Lipton. Promptner: Prompting for named entity recognition. *CoRR*, abs/2305.15444, 2023. doi: 10.48550/ARXIV.2305.15444. URL `https://doi.org/10.48550/arXiv.2305.15444`. 15, 16

[95] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *CoRR*, abs/2305.13168, 2023. URL `https://arxiv.org/abs/2305.13168`. 15, 16

[96] Xingyao Wang, Sha Li, and Heng Ji. Code4struct: Code generation for few-shot event structure prediction. In *ACL (1)*, pages 3640–3663. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.acl-long.202/`. 15, 16

[97] Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. Codeie: Large code generation models are better few-shot information extractors. In *ACL (1)*, pages 15339–15353. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.acl-long.855/`. 15, 16

[98] Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. Codekgc: Code language model for generative knowledge graph construction. *CoRR*, abs/2304.09048, 2023. doi: 10.48550/ARXIV.2304.09048. URL `https://doi.org/10.48550/arXiv.2304.09048`. 15, 16

[99] Yangyi Chen, Xingyao Wang, Manling Li, Derek Hoiem, and Heng Ji. Vistruct: Visual structural knowledge extraction via curriculum guided code-vision representation. In *EMNLP*, pages 13342–13357. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.emnlp-main.824`. 15, 17

[100] Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. Retrieval-augmented code generation for universal information extraction. *CoRR*, abs/2311.02962, 2023. doi: 10.48550/ARXIV.2311.02962. URL `https://doi.org/10.48550/arXiv.2311.02962`. 15, 16

[101] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. Gollie: Annotation guidelines improve zero-shot information-extraction. *CoRR*, abs/2310.03668, 2023. doi: 10.48550/ARXIV.2310.03668. URL `https://doi.org/10.48550/arXiv.2310.03668`. 15, 16

[102] Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. Learning in-context learning for named entity recognition. In *ACL (1)*, pages 13661–13675. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.764. URL `https://doi.org/10.18653/v1/2023.acl-long.764`. 15, 16

[103] Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. How to unleash the power of large language models for few-shot relation extraction? In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 190–200. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.sustainlp-1.13/`. 15, 16

[104] 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 *EMNLP*, pages 3534–3547. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.emnlp-main.214`. 15, 16

[105] Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. Guideline learning for in-context information extraction. In *EMNLP*, pages 15372–15389. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.emnlp-main.950`. 15, 16

[106] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. GPT-NER: named entity recognition via large language models. *CoRR*, abs/2304.10428, 2023. doi: 10.48550/arXiv.2304.10428. URL `https://doi.org/10.48550/arXiv.2304.10428`. 15, 16

[107] Yilmazcan Ozyurt, Stefan Feuerriegel, and Ce Zhang. In-context few-shot relation extraction via pre-trained language models. *arXiv preprint arXiv:2310.11085*, 2023. URL `https://arxiv.org/abs/2310.11085`. 15, 16

[108] Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Self-improving for zero-shot named entity recognition with large language models. *CoRR*, abs/2311.08921, 2023. doi: 10.48550/ARXIV.2311.08921. URL `https://doi.org/10.48550/arXiv.2311.08921`. 15, 16

[109] Hugo Sousa, Nuno Guimarães, Alípio Jorge, and Ricardo Campos. Gpt struct me: Probing gpt models on narrative entity extraction. *CoRR*, abs/2311.08921, 2023. URL `https://arxiv.org/abs/2311.14583`. 15, 16

[110] Chen Ling, Xujiang Zhao, Xuchao Zhang, Yanchi Liu, Wei Cheng, Haoyu Wang, Zhengzhang Chen, Takao Osaki, Katsushi Matsuda, Haifeng Chen, and Liang Zhao. Improving open information extraction with large language models: A study on demonstration uncertainty. *CoRR*, abs/2309.03433, 2023. doi: 10.48550/ARXIV.2309.03433. URL `https://doi.org/10.48550/arXiv.2309.03433`. 15, 16

[111] Jamie P. McCusker. LOKE: linked open knowledge extraction for automated knowledge graph construction. *CoRR*, abs/2311.09366, 2023. doi: 10.48550/ARXIV.2311.09366. URL `https://doi.org/10.48550/arXiv.2311.09366`. 15, 16

[112] Chenran Cai, Qianlong Wang, Bin Liang, Bing Qin, Min Yang, Kam-Fai Wong, and Ruifeng Xu. In-context learning for few-shot multimodal named entity recognition. In *EMNLP (Findings)*, pages 2969–2979. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.findings-emnlp.196`. 15, 16

[113] Jinyuan Li, Han Li, Zhuo Pan, Di Sun, Jiahao Wang, Wenkun Zhang, and Gang Pan. Prompting chatgpt in MNER: enhanced multimodal named entity recognition with auxiliary refined knowledge. In *EMNLP (Findings)*, pages 2787–2802. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.findings-emnlp.184`. 15, 16

[114] Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.67. URL `https://aclanthology.org/2022.findings-acl.67`. 15, 17

[115] Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *ACL (Findings)*, pages 794–812. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.50. URL `https://doi.org/10.18653/v1/2023.findings-acl.50`. 15, 17

[116] 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 *EMNLP*, pages 10030–10051. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.emnlp-main.620`. 15

[117] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. Universalner: Targeted distillation from large language models for open named entity recognition. In *ICLR*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=r65xfUb76p`. 15

[118] 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, 2023. URL `https://doi.org/10.48550/arXiv.2304.08085`. 15, 17

[119] Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. Zero-shot temporal relation extraction with chatgpt. In *BioNLP@ACL*, pages 92–102. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.BIONLP-1.7. URL `https://doi.org/10.18653/v1/2023.bionlp-1.7`. 16

[120] Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Empirical study of zero-shot NER with chatgpt. In *EMNLP*,

pages 7935–7956. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.493. 16

[121] Xilai Ma, Jing Li, and Min Zhang. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In *EMNLP (Findings)*, pages 2334–2352. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-emnlp.153. 16

[122] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022. doi: 10.48550/ARXIV.2210.11416. URL https://doi.org/10.48550/arXiv.2210.11416. 17, 71

[123] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.26. URL https://aclanthology.org/2022.acl-long.26. 17

[124] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124. 18

[125] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

[126] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026. 18

[127] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL `https://aclanthology.org/P19-1346`. 18

[128] Hongjin SU, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=ykuc5q381b`. 18

[129] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.365. URL `https://aclanthology.org/2021.naacl-main.365`. 18

[130] Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. AxCell: Automatic extraction of results from machine learning papers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.692. URL `https://aclanthology.org/2020.emnlp-main.692`. 18, 20

[131] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.254. URL `https://aclanthology.org/2021.acl-long.254`. 18

[132] Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-main.421`. 18

[133] Samira Ebrahimi Kahou, Adam Atkinson, Vincent Michalski, Ákos Kádár, Adam Trischler, and Yoshua Bengio. FigureQA: An annotated figure dataset for visual reasoning, 2018. URL https://openreview.net/forum?id=SyunbfbAb. 19

[134] Daesik Kim, Seonhoon Kim, and Nojun Kwak. Textbook question answering with multi-modal context graph understanding and self-supervised open-set comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3568–3584, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1347. URL https://aclanthology.org/P19-1347.

[135] Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 19

[136] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12), December 2024. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6. URL http://dx.doi.org/10.1007/s11432-024-4235-6. 19

[137] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VI*, page 216–233, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72657-6. doi: 10.1007/978-3-031-72658-3_13. URL https://doi.org/10.1007/978-3-031-72658-3_13. 19, 89, 98

[138] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench: Benchmarking multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13299–13308, 2024. doi: 10.1109/CVPR52733.2024.01263. 19

[139] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM, 2020. URL https://dl.acm.org/doi/10.1145/3394486.3403172. 20

[140] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong

Zhou. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.201. URL `https://aclanthology.org/2021.acl-long.201`.

[141] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548112. URL `https://doi.org/10.1145/3503161.3548112`. 20

[142] Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. DocLLM: A layout-aware generative language model for multimodal document 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 8529–8548, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.463. URL `https://aclanthology.org/2024.acl-long.463/`. 20

[143] Łukasz Borchmann, Michał Pietruszka, Wojciech Jaśkowski, Dawid Jurkiewicz, Piotr Halama, Paweł Józiak, Łukasz Garncarek, Paweł Liskowski, Karolina Szyndler, Andrzej Gretkowski, Julita Ołtusek, Gabriela Nowakowska, Artur Zawłocki, Łukasz Duhr, Paweł Dyda, and Michał Turski. Arctic-tilt. business document understanding at sub-billion scale, 2024. URL `https://arxiv.org/abs/2408.04632`. 20

[144] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022. 20

[145] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Chao-Yue Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19254–19264, 2022. URL `https://api.semanticscholar.org/CorpusID:254275326`. 20

[146] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. 20

[147] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=w0H2xGHlkw`. 20

[148] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, 2024. URL `https://arxiv.org/abs/2404.16821`.

[149] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL `https://arxiv.org/abs/2409.12191`. 21, 113

[150] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Zhe Chen, xinyue zhang, Wei Li, Li Jingwen, Wenhai Wang, Kai Chen, Conghui He, Xingcheng ZHANG, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. InternLM-XComposer2-4KHD: A pioneering large vision-language model handling resolutions from 336 pixels to 4k HD. 2024. URL `https://openreview.net/forum?id=nRp0XhTf61`. 84, 93

[151] Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mPLUG-owl3: Towards long image-sequence understanding in multi-modal large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=pr37sbuhVa`. 20

[152] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL `https://aclanthology.org/2022.findings-acl.177`. 20, 84, 85, 86

[153] Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. Towards complex document understanding by discrete reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4857–4866, 2022. 84

[154] Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. VisualWebBench: How far have multimodal llms evolved in web page understanding and grounding?, 2024. 20

[155] Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C.V. Jawahar. Infographicvqa. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2582–2591, 2021. URL `https://api.semanticscholar.org/CorpusID:233394125`. 20, 84, 105

[156] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multi-page docvqa, 2023. 20, 84

[157] Jordy Van Landeghem, Rubèn Pérez Tito, Łukasz Borchmann, Michal Pietruszka, Pawel J'oziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Ackaert, Ernest Valveny, Matthew B. Blaschko, Sien Moens, and Tomasz Stanislawek. Document understanding dataset and evaluation (DUDE). In *ICCV*, 2023. 20, 85, 86

[158] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. SlideVQA: A dataset for document visual question answering on multiple images. In *AAAI*, 2023. 20, 84, 86

[159] Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. FinanceBench: A new benchmark for financial question answering, 2023. 20, 85, 86

[160] Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-main.823`. 21

[161] Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-eval: Instituting standardized evaluation for long context language models. 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 14388–14411, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.776. URL `https://aclanthology.org/2024.acl-long.776/`.

[162] 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/`.

[163] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens, 2024. 21

[164] Szymon Tworkowski, Konrad Staniszewski, Mikolaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Milo's. Focused transformer: Contrastive training for context scaling. *ArXiv preprint*, abs/2307.03170, 2023. URL https://arxiv.org/abs/2307.03170. 21

[165] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. LongloRA: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6PmJoRfdaK.

[166] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

[167] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. LongAlign: A recipe for long context alignment of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.74. URL https://aclanthology.org/2024.findings-emnlp.74/. 21, 93

[168] Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. Milebench: Benchmarking mllms in long context. *ArXiv preprint*, abs/2404.18532, 2024. URL https://arxiv.org/abs/2404.18532. 21

[169] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning, 2024.

[170] Yujie Lu, Xiujun Li, Tsu-Jui Fu, Miguel Eckstein, and William Yang Wang. From text to pixel: Advancing long-context understanding in mllms, 2024. 21

[171] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997. 21

[172] 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 *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. 21

[173] 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. 21

[174] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726. 21, 113

[175] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents, 2024. URL https://arxiv.org/abs/2410.10594. 21, 103, 112

[176] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Gme: Improving universal multimodal retrieval by multimodal llms, 2024. URL https://arxiv.org/abs/2412.16855. 21

[177] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *ArXiv*, abs/2202.07800, 2022. URL https://api.semanticscholar.org/CorpusID:246867285. 23

[178] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, 2024. URL https://api.semanticscholar.org/CorpusID:268358224. 102, 104, 110

[179] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction, 2024. URL https://arxiv.org/abs/2410.17247.

[180] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference, 2025. URL https://arxiv.org/abs/2410.04417. 23

[181] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 620–640. Springer, 2022. 23

[182] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu. 110

[183] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm, 2024. 23

[184] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models, 2024. URL https://arxiv.org/abs/2403.15388. 23

[185] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models, 2024. URL https://arxiv.org/abs/2412.04467. 23, 104, 110

[186] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.272. URL https://aclanthology.org/2022.naacl-main.272. 24

[187] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57. 24

[188] Benjamin Clavié, Antoine Chaffin, and Griffin Adams. Reducing the footprint of multi-vector retrieval with minimal performance impact via token pooling, 2024. URL https://arxiv.org/abs/2409.14683. 24, 102, 107

[189] Tony Wu. Token Pooling for Colpali. 2024. URL https://x.com/tonywu_71/status/1826238532241010721. 24

[190] Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. Seed-based event trigger labeling: How far can event descriptions get us? In *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 372–376, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2061. URL https://aclanthology.org/P15-2061. 28

[191] Haoruo Peng, Yangqiu Song, and Dan Roth. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1038. URL https://aclanthology.org/D16-1038. 28

[192] Hongming Zhang, Haoyu Wang, and Dan Roth. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.114. URL https://aclanthology.org/2021.findings-acl.114. 28

[193] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Honey or poison? solving the trigger curse in few-shot event detection via causal intervention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8078–8088, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.637. URL https://aclanthology.org/2021.emnlp-main.637. 28

[194] Pengfei Yu, Zixuan Zhang, Clare Voss, Jonathan May, and Heng Ji. Building an event extractor with only a few examples. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 102–109, Hybrid, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deeplo-1.11. URL https://aclanthology.org/2022.deeplo-1.11. 28

[195] Hongming Zhang, Wenlin Yao, and Dong Yu. Efficient zero-shot event extraction with context-definition alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7169–7179, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.531. 28

[196] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf. 29, 54, 65

[197] Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.129. URL `https://aclanthology.org/2020.emnlp-main.129`. 29, 65

[198] Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-0812. URL `https://aclanthology.org/W15-0812`. 29, 65

[199] Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.185. URL `https://aclanthology.org/2021.naacl-main.185`. 30

[200] 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, 2019. URL `https://arxiv.org/abs/1907.11692`. 31

[201] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001. 36, 39

[202] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. ICML'20, 2020. 41

[203] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. doi: 10.1109/CVPR42600.2020.00975. 41

[204] Beth M. Sundheim. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*, 1992. URL `https://aclanthology.org/M92-1001`. 45

[205] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.161. URL `https://aclanthology.org/2021.findings-acl.161`. 48

[206] Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *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`. 49

[207] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too, 2021. 49

[208] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Proceedings of ECCV*, 2020. 52

[209] Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6298–6308, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.492. URL `https://aclanthology.org/2021.acl-long.492`. 52

[210] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 52

[211] Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.718. URL `https://aclanthology.org/2020.acl-main.718`. 54, 65

[212] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1585. URL `https://aclanthology.org/D19-1585`. 54

[213] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 7999–8009, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.713. URL https://aclanthology.org/2020.acl-main.713. 55, 56

[214] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In *International Conference for Learning Representation (ICLR 2023)*, 2023. 63

[215] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-cqvvvb-NkI. 63

[216] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, 2022. 63

[217] Cheng Qian, Chi Han, Yi Fung, Yujia Qin, Zhiyuan Liu, and Heng Ji. CREATOR: Tool creation for disentangling abstract and concrete reasoning of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6922–6939, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.462. URL https://aclanthology.org/2023.findings-emnlp.462/. 63

[218] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.82. 63

[219] Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.248. URL https://aclanthology.org/2023.findings-emnlp.248/. 63

[220] 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/`. 64

[221] 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`. 65

[222] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 2013. 65

[223] Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.248. URL `https://aclanthology.org/2021.acl-long.248`. 65

[224] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *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`. 65

[225] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.142. URL `https://aclanthology.org/2020.acl-main.142`. 65

[226] Xingyao Wang, Sha Li, and Heng Ji. Code4Struct: Code generation for few-shot event structure prediction. 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 3640–3663, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.202. URL `https://aclanthology.org/2023.acl-long.202/`. 66

[227] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155. 67

[228] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. URL https://vicuna.lmsys.org. 67

[229] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10. 67

[230] Hongjin SU, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=qY1hlv7gwg. 67

[231] 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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556. 70

[232] Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.50. URL https://aclanthology.org/2023.findings-acl.50/. 71

[233] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit

Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-main.340`. 71

[234] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022. 75

[235] Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding, 2024. 84, 93

[236] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *ArXiv preprint*, abs/2404.16821, 2024. URL `https://arxiv.org/abs/2404.16821`. 84, 93

[237] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024. 91, 98

[238] Ray Smith. An overview of the tesseract ocr engine. In *ICDAR*, 2007. 92, 93

[239] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. 93

[240] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. 93

[241] Qwen Team. Introducing qwen1.5, 2024. URL `https://qwenlm.github.io/blog/qwen1.5/`. 93

[242] DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model, 2024. 93

[243] Anthropic. Introducing the next generation of claude, 2024. URL https://www.anthropic.com/news/claude-3-family. 93

[244] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. 93

[245] Open AI. Hello gpt-4o, 2024. URL https://openai.com/index/hello-gpt-4o. 93

[246] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. DeepSeek-VL: towards real-world vision-language understanding. *ArXiv preprint*, abs/2403.05525, 2024. URL https://arxiv.org/abs/2403.05525. 93

[247] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024. 93

[248] Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. RLAIF-V: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *ArXiv preprint*, abs/2405.17220, 2024. URL https://arxiv.org/abs/2405.17220. 93

[249] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. LLaVA-UHD: an lmm perceiving any aspect ratio and high-resolution images. *ArXiv preprint*, abs/2403.11703, 2024. URL https://arxiv.org/abs/2403.11703. 93

[250] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *ArXiv preprint*, abs/2308.12966, 2023. URL https://arxiv.org/abs/2308.12966. 93

[251] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *ArXiv preprint*, abs/2311.06607, 2023. URL https://arxiv.org/abs/2311.06607. 93

[252] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, Jiazheng Xu, Xu Bin, Juanzi Li, Jie Tang, and Ming Ding. CogVLM: Visual expert for large language models, 2024. URL https://openreview.net/forum?id=c72vop46KY. 93

[253] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, et al. Generative multimodal models are in-context learners. *ArXiv preprint*, abs/2312.13286, 2023. URL https://arxiv.org/abs/2312.13286. 93

[254] Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding, 2024. URL https://arxiv.org/abs/2411.04952. 102

[255] Jian Chen, Ruiyi Zhang, Yufan Zhou, Tong Yu, Franck Dernoncourt, Jiuxiang Gu, Ryan A. Rossi, Changyou Chen, and Tong Sun. LoRA-contextualizing adaptation of large multimodal models for multi-page document understanding. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FDaHjwInXO. 102

[256] Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1025. URL https://aclanthology.org/P16-1025. 116

[257] Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. CLEVE: Contrastive Pre-training for Event Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6283–6297, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.491. URL https://aclanthology.org/2021.acl-long.491.

[258] Jiaming Shen, Yunyi Zhang, Heng Ji, and Jiawei Han. Corpus-based open-domain event type induction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5427–5440, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.441. URL https://aclanthology.org/2021.emnlp-main.441. 116

[259] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-2322. 117

[260] George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. URL https://aclanthology.org/H94-1111. 117

[261] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational*

Linguistics, Volume 1, pages 86–90, Montreal, Quebec, Canada, August 1998. Association for Computational Linguistics. doi: 10.3115/980845.980860. URL https://aclanthology.org/P98-1013. 117

[262] Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pszewhybU9. 117

[263] Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. Financebench: A new benchmark for financial question answering, 2023. URL https://arxiv.org/abs/2311.11944. 117

[264] DeepSeek-AI Team. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948. 118

[265] Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599. 118

[266] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=tN61DTr4Ed. 119

[267] Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F. Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. Coderag-bench: Can retrieval augment code generation?, 2024. URL https://arxiv.org/abs/2406.14497. 119

[268] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *ArXiv*, abs/2306.06070, 2023. URL https://api.semanticscholar.org/CorpusID:259129428. 119