

LLMAEL: Large Language Models are Good Context Augmenters for Entity Linking

Amy Xin*

Department of Computer Science and
Technology, Beijing National
Research Center for Information
Science and Technology
Tsinghua University
Beijing, China
xin-x25@mails.tsinghua.edu.cn

Yunjia Qi*

Department of Computer Science and
Technology, Beijing National
Research Center for Information
Science and Technology
Tsinghua University
Beijing, China
qyj23@mails.tsinghua.edu.cn

Zijun Yao

Department of Computer Science and
Technology, Beijing National
Research Center for Information
Science and Technology
Tsinghua University
Beijing, China
yaozj20@mails.tsinghua.edu.cn

Fangwei Zhu

Beijing University
Beijing, China
zhufangwei2022@stu.pku.edu.cn

Kaisheng Zeng

Tsinghua University
Beijing, China
zks19@mails.tsinghua.edu.cn

Bin Xu

Tsinghua University
Beijing, China
xubin@tsinghua.edu.cn

Lei Hou[†]

Department of Computer Science and
Technology, Beijing National
Research Center for Information
Science and Technology
Tsinghua University
Beijing, China
houlei@tsinghua.edu.cn

Juanzi Li[†]

Department of Computer Science and
Technology, Beijing National
Research Center for Information
Science and Technology
Tsinghua University
Beijing, China
lijuanzi@tsinghua.edu.cn

Abstract

Specialized entity linking (EL) models are well-trained at mapping mentions to unique knowledge base (KB) entities according to a given context. However, specialized EL models struggle to disambiguate long-tail entities due to their limited training data. Meanwhile, extensively pre-trained large language models (LLMs) possess broader knowledge of uncommon entities. Yet, with a lack of specialized EL training, LLMs frequently fail to generate accurate KB entity names, limiting their standalone effectiveness in EL. With the observation that LLMs are more adept at context generation instead of EL execution, we introduce LLM-Augmented Entity Linking (LLMAEL), the first framework to enhance specialized EL models with LLM data augmentation. LLMAEL leverages off-the-shelf, tuning-free LLMs as context augmenters, generating entity descriptions to serve as additional input for specialized EL models. Experiments show that LLMAEL sets new state-of-the-art results across 6 widely adopted EL benchmarks: compared to prior methods that integrate tuning-free LLMs into EL, LLMAEL achieves

an absolute 8.9% gain in EL accuracy. We release our code and datasets¹.

CCS Concepts

• **Software and its engineering** → *Entity relationship modeling*; • **Computing methodologies** → **Natural language generation**.

Keywords

Entity Linking, Entity Disambiguation, Large Language Models, Knowledge Graphs, Data Augmentation

ACM Reference Format:

Amy Xin, Yunjia Qi, Zijun Yao, Fangwei Zhu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. 2025. LLMAEL: Large Language Models are Good Context Augmenters for Entity Linking. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*, November 10–14, 2025, Seoul, Republic of Korea. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3746252.3761156>

1 Introduction

Entity linking (EL) is the task of establishing connections between mentions in textual contexts and unique entities in a knowledge base (KB). It plays an important role in many applications that require semantic understanding, including question answering (QA) [28–30], dialogue generation [8, 20], and recommendation systems [3, 25]. Effective entity linking is crucial to the reliable performance of such downstream tasks. For example, when a QA

*Equal Contribution.

[†]Corresponding Authors.



This work is licensed under a Creative Commons Attribution 4.0 International License. *CIKM '25, Seoul, Republic of Korea*

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2040-6/2025/11

<https://doi.org/10.1145/3746252.3761156>

¹<https://github.com/THU-KEG/LLMAEL>

system answers the question “When was Michael Jordan born?”, the system must correctly disambiguate whether the mention “Michael Jordan” refers to the basketball player *Michael Jordan* or the UCB professor *Michael Jordan* in order to output accurate results.

However, entity linking is still a challenging task as it requires two distinct capabilities: (a) Task Specification, which encompasses a thorough understanding of the entity linking task and the precise requirement for its output format, and (b) Entity Knowledge, which involves the possession of substantial knowledge about the target entity. Trained specifically for entity linking, specialized EL models [1, 5, 26] excel in task specification, capable of producing accurate entity names that exactly satisfy the format requirement of the EL task. Meanwhile, extensively pre-trained large language models (LLMs) [4, 23] are natural repositories of expansive world knowledge, possessing vast parametric knowledge pertinent to any given entity.

As LLMs have gained widespread recognition and accessibility in recent years, several prior studies have explored the integration of LLMs into the entity linking task. Existing methods can be grouped into two categories: (1) *Fine-tuning-based methods*, which involve training LLMs for entity linking through supervised fine-tuning [27]; (2) *Prompting-based methods*, which employ in-context learning to prompt LLMs for entity linking [9, 22]. However, both approaches directly employ LLMs as *EL executors*, leading to drawbacks for each method. First, fine-tuning-based methods, despite yielding promising results, require significant training expenses to tune LLMs’ massive parameters. Conversely, prompting-based methods are more cost-efficient, but yield less accurate results due to LLMs’ limitations in EL task specification. Hence, we ask the following question: *How can we integrate LLMs into the entity linking task to effectively balance computational cost and performance?*

Motivated by this challenge, we introduce LLM-Augmented Entity Linking (LLMAEL), the first entity linking framework to employ LLMs as data augmenters for specialized EL models. Unlike previous approaches that directly utilize LLMs as *EL executors*, LLMAEL employs LLMs as *EL context augmenters*, providing specialized EL models with supplemental context to aid entity linking execution. LLMAEL effectively addresses the limitations of previous works, particularly the trade-off between computational cost and performance. In terms of cost, LLMAEL interacts with LLMs via prompting, offering a lightweight solution free from expensive LLM fine-tuning. In terms of performance, LLMAEL achieves promising results by leveraging specialized EL models for final entity linking execution. Overall, LLMAEL successfully combines the extensive entity knowledge of LLMs with the task-specific capabilities of specialized EL models, delivering an LLM-enhanced EL solution that effectively balances computational efficiency and performance accuracy.

Our paper’s contributions are as follows:

- (1) We propose LLMAEL, the first entity linking framework that leverages LLMs as context augmenters for specialized EL models. We show that LLMAEL is a flexible framework with multiple advanced configurations that further enhance its performance.
- (2) LLMAEL yields promising results: compared to prior methods that integrate tuning-free LLMs into the EL task, LLMAEL yields absolute accuracy gains as high as 8.9%; compared to

only using specialized EL models, LLMAEL achieves absolute average accuracy improvements of 1.2%.

- (3) We further explore alternative means of integrating tuning-free LLMs into the entity link process, and demonstrate that LLMAEL is the most effective approach.

2 Preliminaries and Related Work

2.1 Task Definition

Entity Linking (EL) is the task of mapping mentions from a given context to KB entities. Formally, the knowledge base G consists of the set of entities $\{e\}$ that are unique objects in the real world. The input of entity linking is a textual context c , embedded with multiple entity mentions, denoted as $c = \dots t_1 || m_1 || t_2 || m_2 || t_3 \dots$, where t_i are textual spans and m_i are entity mentions. The goal of entity linking is to obtain a correct list of mention-entity pairs $\{(m_i, e_i)\}_{i \in [1, k]}$.

2.2 Related Work

Specialized Entity Linking Models. It has been a long-standing goal to develop models specialized for entity linking. The most widely adopted solution to build specialized EL models is a two-stage architecture [21], which divides EL into two sequential phases: entity candidate generation and entity re-ranking. Most models approach the entity candidate generation phase as a retrieval problem, aligning mentions to entities according to BM25 ranking[15] or embedding similarity scores[13, 26]. With the development of generative language models, it has also become possible to treat entity candidate generation as a text generation task [5], training the model to generate unique entity names in the knowledge base directly based on the contextual information. Furthermore, most recent works prove that concept information about mentions is useful for EL, thus fine-grained entity typing is also integrated as part of the pipeline and has been applied to numerous models [1, 17]. This suggests that augmenting mentions with additional information facilitates the entity linking process, leaving space for further research on EL context augmentation.

LLMs as Executors for Downstream Tasks. In-context learning, or few-shot prompting, is a prevalent strategy that enables LLMs to execute downstream tasks without the need of LLM fine-tuning. With the outstanding accomplishments of LLMs like GPT-3 [4] and LLaMA2 [23], LLM in-context learning has achieved impressive results in downstream task execution, such as question answering, summarization, and machine translation. However, research shows that LLM in-context learning still falls short when executing *specification-heavy tasks* [18], which are tasks with complicated and extensive task specifications (e.g., entity linking). For these tasks, LLM in-context learning yields significantly inferior results compared to specialized models. Therefore, relying on LLM in-context learning for direct entity linking execution is not the optimal approach. This underscores the need for more effective means to integrate LLMs into the entity linking process.

LLMs as Context Augmenters for Downstream Tasks. LLMs are primarily designed for text generation, which is their strongest advantage. Numerous studies have shown that LLM-generated contexts present outstanding qualities, outperforming contexts obtained

from information retrieval methods [6, 31]. Furthermore, compared to retrieved contexts, LLM-generated contexts contribute to better downstream task performance [6]. With such insights, a bright solution is to leverage LLMs as *context augmenters* instead of task executors: to generate supplemental context for specialized models as additional input. For instance, Liu et al. [14] demonstrate how LLM context augmentation can enhance commonsense reasoning, achieving state-of-the-art results on multiple reasoning tasks. Similarly, Balkus and Yan [2] show that GPT-3 augmented data can improve specialized text classification models, resulting in higher consistent accuracy on unseen examples. However, there have been no attempts on employing LLM context augmentation to enhance specialized entity linking models.

LLMs for Entity Linking. Several prior works have integrated LLMs into the entity linking task. Yet, all works employ LLMs as direct task executors for entity linking. The methodologies of these works can be categorized into two lines of approaches:

Fine-tuning-based methods. This approach leverages LLMs by directly fine-tuning them specifically for the entity linking task. At test time, the fine-tuned LLM serves as the entity linking model to directly generate entity predictions. A representative example is InsGENEL [27], which introduces a sequence-to-sequence training objective for EL and applies instruction fine-tuning to enable LLMs to execute the EL task. However, while effective, the fine-tuning approach is computationally expensive, making it a costly solution.

Prompting-based methods. In contrast to fine-tuning, prompting-based methods employ LLMs as entity linkers through in-context learning. These methods design carefully crafted prompts for EL, which are then inputted into the LLM to generate entity predictions. For instance, ChatEL [9] presents a three-step framework to prompt LLMs for EL. Similarly, GEMEL [22] applies prompting-based techniques to multimodal EL tasks and incorporates a constrained beam search strategy to refine entity generation. However, these methods face challenges due to the inherent limitations of LLMs in accurately generating unique entity names, as they lack specialized training on the task. Consequently, prompting-based methods often yield inferior results compared to specialized EL models like ReFinED [1] and GENRE [5], which are both more effective and cost-efficient.

Comparing LLMAEL to Previous Works. LLMAEL is the first approach to enhance entity linking by employing LLM context augmentation over specialized EL models. LLMAEL could be compared to two categories of existing work: (1) methods that use LLM context augmentation to enhance downstream tasks, and (2) methods that employ LLMs for entity linking.

For category (1), as detailed in the *LLMs as Context Augmenters for Downstream Tasks* paragraph (2.2), there are methods that successfully employ LLM context augmentation to enhance specialized models in other tasks, such as commonsense reasoning and text classification. However, there has been no exploration into how LLM context augmentation can be used to enhance specialized entity linking models, and LLMAEL is the first attempt.

Similarly, for category (2), as detailed in the *LLMs for Entity Linking* paragraph (2.2), there are works that leverage LLMs for entity linking, namely both *fine-tuning-based methods* and *prompting-based methods*. However, all these methods directly employ LLMs as EL executors, which leads to either large training costs in the

fine-tuning scenario or inferior performance in the tuning-free scenario. In contrast, LLMAEL is the first method to employ tuning-free LLMs as *EL context augmenters* for specialized EL models, effectively addressing the cost-performance trade-off.

To sum up, LLMAEL stands as a novel approach that explores the effective collaboration of general LLMs with specialized models under the task of entity linking, combining the strengths of both model types at a low cost.

3 Methodology

LLMAEL is the first plug-and-play framework to employ LLMs as context augmenters for specialized EL models. Figure 1 illustrates the overall system architecture of LLMAEL. We mark the traditional entity linking process in pink and our method in yellow. Mentions that need to be executed by entity linking are marked with blue. As presented by Figure 1, LLMAEL mainly includes three building blocks. (1) **Context augmentation** is the most basic element for LLMAEL, which elicits LLMs to generate enriched context with more information for entity linking. (2) **Data fusion** designs multiple strategies to integrate the LLM-generated content with the original context, aiming to improve diverse off-the-shelf EL models. (3) **EL execution** finally conducts the entity linking task.

3.1 Context Augmentation

In the Context Augmentation stage, we employ tuning-free LLMs to generate supplementary context for a given mention. Formally, we denote LLM context augmentation as the function

$$c' = \text{LLM}(p, c, m_i)$$

where p is a specially designed prompt to instruct LLMs for context generation, c is the original context for the i^{th} mention m_i , and c' is the augmented context generated by the LLM.

We implement LLM context augmentation through in-context learning [4] with a three-shot prompt. Our prompt comprises two main parts: (1) task specification instructions on context augmentation, and (2) in-prompt demonstrations. For task specification, we use the following template to prompt LLMs:

Consider the following text.
Text: [CONTEXT]
Please provide me more descriptive information about [MENTION] from the text above.
Make sure to include [MENTION] in your description.

where [CONTEXT] and [MENTION] are placeholders to be filled before feeding into LLMs. It is worth noting that our instruction requires LLMs to recite the entity again in the generated context, which provides flexibility for later data fusion. For in-prompt demonstrations, we bootstrap examples via zero-shot prompting. To ensure the quality of these demonstrations, we first generate a sufficiently large number of LLM responses via zero-shot prompting, and then manually filter out high-quality samples. The final prompt's exemplars are then selected from this high-quality pool.

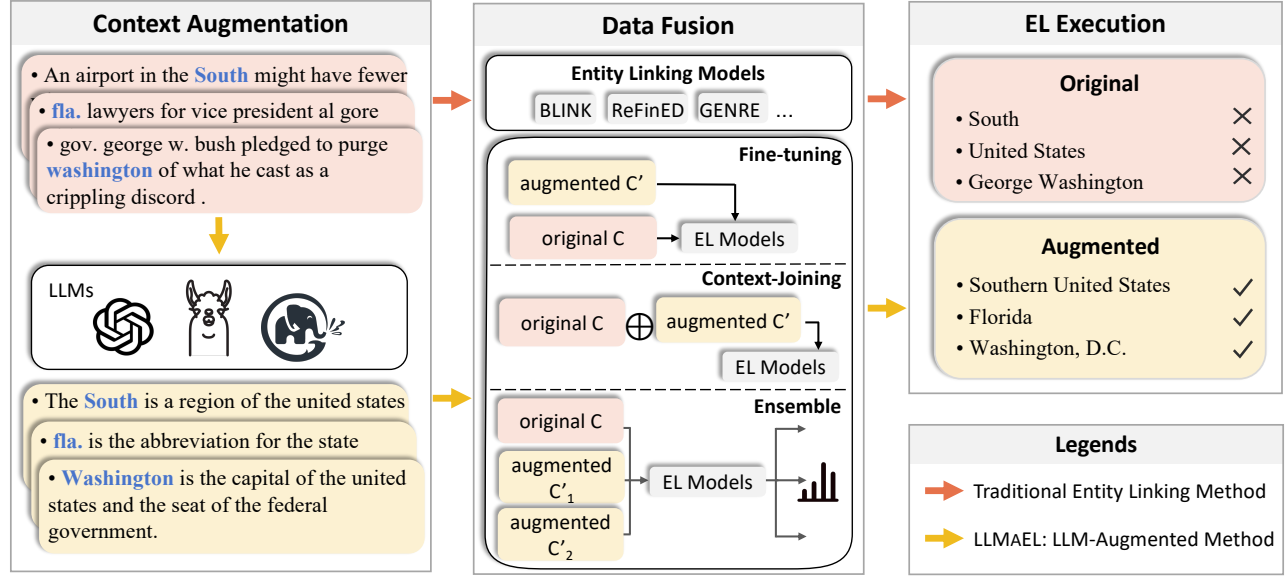


Figure 1: Proposed methodology architecture.

Table 1: Five possible context-joining strategies for LLMAEL.

ID	Context Order	Mention Offset
0	LLM-only	LLM
1	LLM + Original	LLM
2	LLM + Original	Original
3	Original + LLM	LLM
4	Original + LLM	Original

3.2 Data Fusion

The Data Fusion stage designs multiple strategies to integrate LLM-augmented context c' into the EL process. In particular, *context-joining strategies* directly augment c' into the original context c ; *EL model fine-tuning* infuses the knowledge in c' into the EL model's parameters via lightweight fine-tuning; and *ensemble* combines multiple LLM-augmented contexts through output majority voting.

Context-Joining Strategies. A critical consideration of LLMAEL's implementation is determining how to integrate the LLM-augmented context c' with the original context c from the text source. The most straightforward approach is to concatenate c' and c into a whole passage, then input the combined context into off-the-shelf EL models. This approach raises two key design questions:

- (1) **Context Order:** In what sequence should c and c' be concatenated?
- (2) **Mention Offset:** Since the mention appears in both c and c' , which mention span should be selected as the target span for the EL models?

To address these questions, LLMAEL considers five potential strategies for joining the original and LLM-generated contexts, as

outlined in Table 1. These strategies enumerate different arrangements of c and c' in the final augmented context provided to the EL model. In Table 1, *Context Order* specifies the sequential arrangement of the original and LLM-generated contexts, while *Mention Offset* determines which context's mention span is used by the EL model. Specifically:

- **Strategy 0** substitutes c' directly for c , discarding the original context.
- **Strategies 1–4** explore all four possible combinations of context order and mention offset.

Our experiments reveal that the optimal joining strategy varies across different EL models. Consequently, we treat the choice of context-joining strategy as a tunable hyperparameter, enabling users to adapt LLMAEL to various application settings for optimal performance.

EL Model Fine-tuning. A key challenge for LLMAEL is that the style and distribution of the augmented EL context may differ from what EL models are accustomed to, potentially resulting in suboptimal performance. To address this mismatch between the text distributions familiar to EL models and the LLM-augmented contexts, we explore the option of fine-tuning existing EL models. Specifically, we first perform LLM context augmentation on existing EL training datasets as described in 3.1, where an LLM generates mention descriptions for each entry of the training set. Then, the generated descriptions are incorporated into the training set using the optimal context-joining strategy of the selected EL model. Finally, we fine-tune the EL model on this augmented training set to improve its compatibility with the LLM-augmented contexts.

Ensemble. Inaccuracies in LLMAEL's performance may occur when the LLM generates incorrect mention descriptions, misguiding the EL model to output an incorrect entity. Hence, we also expand LLMAEL with ensemble techniques, attempting to improve

LLMAEL’s robustness through diversified sampling. We sample mention descriptions across multiple LLMs and evaluate the diversified samples through both hard-voting and soft-voting classifier methods.

3.3 EL Execution

In the final stage of EL execution, the specialized EL model is employed to output the KB entity name. Compared to directly tasking LLMs to perform entity linking, LLMAEL ensures EL task specification using specialized EL models, while augmenting the EL models with expansive entity knowledge from the LLM-generated context.

4 Experimental Setup

Backbone Models. For our main experiments, we use Llama-3-70b-Instruct as the backbone LLM. We select Llama-3-70b-Instruct for two main reasons. First, as a context augmentation framework, LLMAEL aims to achieve robust performance with open-source LLMs. By leveraging such an open-source LLM, we prioritize contributing to the research community with a solution that is both openly accessible and cost-effective. Second, LLMAEL requires an LLM with exceptional text generation capabilities to meet the demands of EL context augmentation. Llama-3-70b-Instruct currently stands as one of the best-performing open-source LLMs for text generation, making it a natural fit for our experiments. Additionally, our experiments reveal that Llama-3-70b-Instruct delivers comparable, and in some cases superior, performance to proprietary LLMs when applied to LLMAEL, which will be detailed in Section 5.2.1. For all experiments, we set the number of LLM maximum output tokens to 150. We set the temperature of Llama-3-70b-Instruct to 0.01, and adapt default values for all other hyperparameters.

As LLMAEL is a plug-and-play framework for any specialized EL models, we evaluate three most widely adopted EL models as our backbones: BLINK [26], a classical bi-encoder cross-encoder EL model; GENRE [5], an autoregressive generative EL model; and ReFinED [1], an EL model enhanced with entity typing and descriptions. All three models are state-of-the-art for their respective model architectures. Specifically, we use the full BLINK model², the fairseq-AIDA GENRE model³, and the AIDA ReFinED model⁴. To ensure a fair assessment across EL models, we evaluate GENRE without the pregenerated candidate set.

Datasets. We evaluate LLMAEL on six widely adopted entity linking (EL) datasets: AIDA-YAGO2 [12], MSNBC [7], AQUAINT [16], ACE2004 [19], WNED-CWEB [10], and WNED-WIKI [11]. These datasets are selected for their popularity and recognition in the EL field, as well as their ability to cover diverse domains, including news articles, web pages, and encyclopedia corpora.

For each of our three backbone EL models, we first download their official training and testing splits from their respective github repositories^{5,6,7}. These repositories provide standard versions of the datasets required for both the evaluation and training of LLMAEL,

including the test splits for MSNBC, AQUAINT, ACE2004, WNED-CWEB, WNED-WIKI, and the train, dev, and test splits for AIDA-YAGO2. We use our backbone LLM to generate mention descriptions for all eight data splits for the ReFinED model, and just the six test splits for the other two EL models.

Baselines. We compare LLMAEL with two categories of baselines: (1) *LLMs for EL*. First, we evaluate a naive *LLM only* baseline that prompts an LLM to execute the EL task through in-context learning. Specifically, we prompt Llama-3-70b-Instruct with three-shot examples of mention-context pairs and gold entity names, and demand it to output the entity name for the current test case. Second, we compare with ChatEL [9], the state-of-the-art prompting-based framework that uses tuning-free LLMs for entity linking. (2) *Specialized EL models*. We also compare with each of our three backbone EL models in a standalone setting without any LLM data augmentation.

Evaluation Metrics. We use *disambiguation accuracy* as our evaluation metric, defined as the ratio of correctly linked entities to the total number of entities in the dataset:

$$\text{Accuracy} = \frac{\# \text{ Correctly linked entities}}{\text{Total \# of entities in dataset}}$$

For experiments involving an EL model, we first augment the model’s official datasets with LLM-generated context. The augmented datasets are then evaluated using the model’s official testing scripts to ensure consistency with prior benchmarks. For LLM-only baseline experiments that do not incorporate EL models, we manually constructed an evaluation script to compute the LLM’s disambiguation accuracy, where we use the *Exact Match (EM)* measure defined below:

$$\text{Exact Match (EM)} = \begin{cases} 1 & \text{if } \hat{y} = y \\ 0 & \text{otherwise} \end{cases}$$

Where $\hat{y}$ is the LLM’s predicted entity name and y is the ground truth entity name. The measure only outputs 1 if $\hat{y}$ is exactly equal to y , character by character. Then, we divide the number of exact match entities by the total number of entities in the dataset to obtain the final disambiguation accuracy.

Selecting a Unified Context-Joining Strategy. We use the dev split of AIDA-YAGO2 to determine the optimal context-joining strategy. In our main experiments, we consistently report results obtained by Strategy 4 across all three EL models, as it achieves the highest average accuracy for each model on the dev set, ensuring a fair comparison. We hypothesize that Strategy 4 is superior because most EL models are more accustomed to the distribution of original contexts, thus yielding better performance when LLM-generated contexts are appended strictly after the original contexts.

We conduct context-joining strategy experimentation with the following implementation details. For joining strategies that incorporate context combination, contexts are merged using a newline symbol “\n”. In the case of BLINK and ReFinED, contexts are fully combined in the specified order, without any truncation applied. In the case of GENRE, the contexts are first fully combined and then trimmed to the model’s maximum input sequence length.

²BLINK’s full cross-encoder model

³The GENRE model developed using the fairseq toolkit and officially fine-tuned on AIDA-YAGO2

⁴The ReFinED model officially fine-tuned on AIDA-YAGO2

⁵<https://github.com/facebookresearch/BLINK>

⁶<https://github.com/facebookresearch/GENRE>

⁷<https://github.com/amazon-science/ReFinED>

Table 2: Main results of LLMAEL across six EL benchmarks.

Method	AIDA	MSNBC	AQUA	ACE04	CWEB	WIKI	AVG
LLM only	78.37	80.49	73.18	83.27	65.34	64.44	74.18
ChatEL [9]	81.20	83.50	75.70	85.60	66.20	74.50	77.78
BLINK only	82.01	86.23	85.16	86.01	69.11	81.11	81.61
GENRE only	87.92	83.54	84.32	84.82	68.75	83.02	82.06
ReFinED only	92.25	<u>87.10</u>	87.53	<u>87.75</u>	72.96	85.18	85.46
LLMAEL × BLINK	81.94	86.56	85.16	86.01	69.17	81.14	81.66
LLMAEL × GENRE	88.27	85.67	85.14	85.21	70.67	82.95	82.99
LLMAEL × ReFinED	92.38	86.94	<u>88.09</u>	88.14	<u>73.16</u>	<u>85.90</u>	<u>85.76</u>
LLMAEL × ReFinED _{FT}	<u>92.34</u>	88.79	89.06	88.14	75.07	86.62	86.67

Fine-tuning. We select our best-performing EL model ReFinED for model fine-tuning. We use the train and dev splits from the AIDA-YAGO2 dataset as our training and evaluation data. To avoid model over-fitting on AIDA-YAGO2, we leverage ReFinED’s wikipedia model for fine-tuning. Specifically, we first employ Llama-3-70b-Instruct to augment the datasets under the model’s optimal context-joining strategy, then apply the augmented datasets for EL model fine-tuning.

5 Results and Discussions

5.1 Main Results

The results of our main experiments are presented in Table 2. Table 2’s results can be divided into four horizontal sections, ordered from top to bottom. The first section presents results of baselines that only leverage tuning-free LLMs. The second section presents results of standalone EL models. The third section presents results of the vanilla implementation of LLMAEL, representing a straightforward context augmentation approach without any EL model fine-tuning. The final section reports the results of LLMAEL with a fine-tuned ReFinED model, denoted as ReFinED_{FT}. For each dataset, the highest disambiguation accuracy score is highlighted in bold, while the second-highest score is underlined. The results of ChatEL are taken from their official github repository⁸, while all other results are produced by ourselves.

The vanilla implementation of LLMAEL improves the average accuracy score of the six datasets across all three EL models. Vanilla LLMAEL enhances performance on at least five datasets for each EL model, with LLMAEL × GENRE demonstrating an average improvement of 0.93% over its EL model backbone. It is also important to note that, while both incorporating LLMs under a completely tuning-free scenario, vanilla LLMAEL yields significantly better results than the ChatEL baseline (7.68% higher accuracy with LLMAEL × ReFinED), justifying the effectiveness of our method design.

The fine-tuned LLMAEL × ReFinED_{FT} consistently delivers new state-of-the-art results across all six datasets, offering a 1.21% accuracy improvement over the original ReFinED backbone and a 0.91% increase compared to the vanilla LLMAEL × ReFinED. This supports our hypothesis that fine-tuning further enhances performance by

better aligning EL models with the token distributions of LLM-augmented contexts. Compared to ChatEL, LLMAEL × ReFinED_{FT} further yields a performance gain of 8.89%.

Two datasets in particular are worth specific discussion in the LLMAEL × ReFinED_{FT} configuration. First, on the AIDA test set, there is a slight performance reduction from 92.38% to 92.34%. We hypothesize that this is due to potential overfitting to the AIDA dataset distribution during fine-tuning with ReFinED_{FT}, resulting in a minor performance drop. Secondly, ACE2004 results appear capped at 88.14%, both with and without fine-tuning. This suggests that 88.14% may be the maximum achievable accuracy for the ReFinED model on the ACE2004 test set, likely due to inherent limitations of the dataset and the model. On all other datasets, ReFinED_{FT} significantly boosts performance.

5.2 Ablation Studies

In this section, we conduct ablation studies to test the robustness of LLMAEL and to experiment with additional strategies that further enhance performance. In Section 5.2.1, we employ other state-of-the-art LLMs on our method to demonstrate LLMAEL’s robustness across LLM choices, as well as to justify our choice of Llama-3-70b-Instruct as the primary LLM backbone. In Sections 5.2.2 and 5.2.3, we explore additional strategies that further boosts LLMAEL’s performance.

5.2.1 Choosing Among LLMs. As LLMAEL is adaptable to any LLM, in this section, we investigate LLMAEL’s performance across different backbone LLM choices. We compare our primary backbone LLM Llama-3-70b-Instruct to two other widely recognized proprietary LLMs, namely GPT-3.5-Turbo-Instruct and GLM-4. We set the temperature of Llama-3-70b-Instruct and GLM-4 to 0.01 and the temperature of GPT-3.5-Turbo-Instruct to 0, and adapt default values for all other hyperparameters.

The upper half labeled “Single” of Table 3 presents the results of backbone LLM ablation. Again, LLMAEL × ReFinED presents our vanilla context augmentation approach, while ReFinED_{FT} presents LLMAEL with our custom fine-tuned ReFinED EL model. For the vanilla LLMAEL × ReFinED, all three LLMs demonstrate an average performance enhancement. Among them, Llama-3-70b-Instruct

⁸https://github.com/yifding/In_Context_EL

Table 3: Results of LLMAEL across different backbone LLM choices.

	EL Model	LLMAEL LLM backbone(s)	MSNBC	AQUA	WIKI	AVG
Single	ReFinED	-	87.10	87.53	85.18	86.60
		Llama-3-70b-Instruct	86.94	88.09	85.90	86.98
		GPT-3.5-Turbo-Instruct	86.94	88.23	85.60	86.92
		GLM-4	86.94	87.95	85.75	86.88
	ReFinED _{FT}	-	89.40	89.20	85.93	88.18
		Llama-3-70b-Instruct	88.79	89.06	86.62	88.16
		GPT-3.5-Turbo-Instruct	89.40	89.47	86.28	88.38
		GLM-4	89.40	89.20	86.24	88.28
Multi	ReFinED	Hard-voting ensemble	86.94	87.95	85.90	86.93
		Soft-voting ensemble	86.94	87.95	85.75	86.88
	ReFinED _{FT}	Hard-voting ensemble	89.09	89.47	86.62	88.39
		Soft-voting ensemble	89.40	89.34	86.25	88.33

Table 4: Results of vanilla LLMAEL implemented with BLINK under different context-joining strategies.

Method	ID	AVG acc.
BLINK only	-	81.61
LLMAEL × BLINK	4	81.66
LLMAEL × BLINK	1*	84.70

yields the most significant overall improvement, achieving an average accuracy of 86.98%. GPT-3.5-Turbo-Instruct and GLM-4 demonstrate comparable performance, yielding average improvements of at least 0.28%. The fine-tuned ReFinED_{FT} also demonstrates excellent compatibility across various LLMs. With GPT-3.5-Turbo-Instruct, ReFinED_{FT} achieves an average performance of 88.38%, representing a 1.78% improvement over the original context and model. It is also noteworthy that the performance of ReFinED_{FT} also shows a significant enhancement when applied to the original contexts, registering an average performance improvement of 1.58%.

Overall, different LLM backbones contribute only minor performance variations to our method, with a maximum difference of 0.38% in the vanilla setting and 0.22% in the fine-tuned setting. This leads to two key conclusions. First, it confirms that LLMAEL is a robust framework that delivers consistent performance across a variety of LLMs. This allows us considerable flexibility in choosing among various open-source and proprietary LLMs while maintaining the framework’s promising performance. Second, the fact that Llama-3-70b-Instruct achieves the best performance in the tuning-free setting further validates our primary backbone LLM choice. This highlights that our system performs best with open-source LLMs, offering an effective, accessible, and cost-efficient solution.

5.2.2 Model-Specific Context-Joining Strategies. For our main experiments, we used the development set of the AIDA benchmark to select a unified context-joining strategy. However, during our

development set experiment, we observed that different EL models actually show optimal performance with different context-joining strategies. This is because our three selected EL models each possess different model architectures, leading each model to prefer different data distributions during inference.

Namely, BLINK’s optimal context-joining strategy significantly diverges from ReFinED and GENRE. BLINK’s optimal context-joining strategy is strategy 1, as presented in Table 2. The table’s *AVG acc.* column presents the unweighted macro average disambiguation accuracy score of vanilla LLMAEL × BLINK over all 6 test sets. Results show that adopting this model-specific optimal strategy for BLINK leads to a significant performance enhancement of 3.04% in average accuracy. Intriguingly, BLINK’s optimal test-time strategy (strategy 1) has completely opposite parameters as unified strategy 4. We hypothesize that the reliance on AIDA-dev for selecting the optimal joining strategy might be a contributing factor. Given that BLINK, unlike the other two EL models, is not fine-tuned on the AIDA dataset, it may not resonate well with the textual distributions of AIDA datasets. Consequently, BLINK’s performance on the AIDA-dev dataset does not accurately reflect its true preferences and capabilities.

5.2.3 Ensemble. We use both hard-voting and soft-voting classifiers to perform ensemble. The hard-voting classifier is executed by selecting the most frequent outcome among multiple independently-generated results. In instances where multiple results share an equivalent frequency, the result with the highest probability level is selected. Conversely, the soft-voting classifier selects the final answer by aggregating the probabilities of all outcomes.

The *Multi* half of Table 3 illustrates our ensemble results. Both ReFinED and ReFinED_{FT} are improved by the implementation of ensemble techniques. For ReFinED, ensemble using the hard-voting classifier achieves the highest average accuracy of 86.93%. This accuracy score is higher than the score obtained by the soft-voting classifier. This is because the hard-voting classifier is particularly effective when the performance of individual models is diverse. For most datasets, the original ReFinED model yields results that

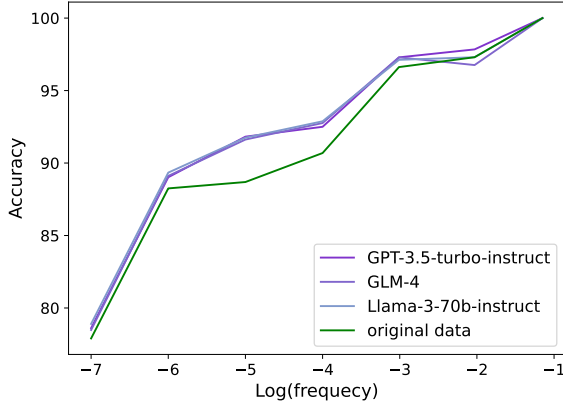


Figure 2: Results of LLMAEL across entities of different frequencies.

are apparently different to the other 3 LLM-enhanced models, contributing to the diversity of model performance. Meanwhile, when the performance of single models is relatively uniform, both ensemble methods—hard and soft-voting classifiers—exhibit comparable effectiveness. This phenomenon is evident in the performance outcomes of the ReFinED_{FT}, where the hard and soft-voting classifiers present equal average accuracies. Moreover, the hard-voting classifier under ReFinED_{FT} also achieved the best average performance of 88.39%.

5.3 Discussions

We delve deeper into LLMAEL by examining the following two discussion questions.

5.3.1 Does LLM-Augmented Data Improve EL Performance Over Long-Tail Entities? In this section, we examine LLMAEL’s effectiveness across entities of varying frequencies. EL models typically perform well on high-frequency entities but struggle with low-frequency ones due to limited training data, whereas LLMs encode broader entity knowledge. We hypothesize that LLMAEL’s gains stem from improving long-tail entity linking.

To test this, we evaluate LLMAEL on MSNBC, AQUAINT, ACE04, and WNED-WIKI. Each entity is assigned a Wikidata5M [24] PageRank score as its frequency, normalized on a base-10 logarithmic scale and grouped into seven buckets.

Figure 2 shows bucket accuracies: the green line is the baseline ReFinED, while purple lines show ReFinED_{FT} trained with LLM-augmented data. Results confirm that LLMAEL substantially improves mid- to low-frequency entities (10^{-6} – 10^{-2}) and refines performance on extremely rare ones (10^{-7} – 10^{-6}). This supports our hypothesis that LLM augmentation benefits long-tail entities, while also boosting mid-tail performance by providing cleaner, condensed mention contexts.

5.3.2 Is There a Better Way to Leverage LLMs for EL? LLMAEL investigates the effective collaboration between LLMs and specialized EL models by LLM context augmentation. In this section, we

Table 5: Results of various LLM-for-EL approaches.

Method	AIDA	MSNBC	AQUA	ACE04
BLINK only	82.01	86.23	85.16	86.01
LLM only	78.37	80.49	73.18	83.27
Re-rank-100	70.95	82.01	73.18	74.32
Re-rank-10	73.24	80.18	73.45	82.88

explore another possible way to combine LLMs with EL models: using specialized EL models for candidate entity retrieval, and then leveraging LLMs to conduct entity re-ranking.

Considering that many EL models, such as BLINK, operate by first retrieving candidate entities and then re-ranking them, a practical approach to combine LLMs and specialized EL models is to use EL models for candidate entity retrieval and LLMs for entity re-ranking. We use BLINK’s bi-encoder for candidate retrieval and Llama-3-70b-Instruct for re-ranking. We evaluate two entity re-ranking settings:

- **Re-rank-100:** Extract the top 100 candidate entities from BLINK’s bi-encoder and let the LLM select the final entity.
- **Re-rank-10:** Extract the top 10 candidate entities of BLINK’s bi-encoder and augment each candidate with its Wikipedia abstract. The LLM selects the final entity using the abstracts as supplementary information.

As shown in Table 5, using LLMs for entity re-ranking does not improve EL performance. We observe two primary reasons. First, the presence of similar candidate names confuses the LLM. Unlike demanding LLMs to directly generate entity names for mentions, asking LLMs to perform re-ranking requires them to discern the subtle distinctions among candidates. As highlighted by Peng et al. [18], LLMs struggle to understand and distinguish complex contexts, leading to diminished performance. Secondly, presenting the LLM with multiple candidates often causes it to spread its focus across the entire context rather than concentrating on the specific mention. This results in the LLM prioritizing information that is distant and unrelated to the mention. The suboptimal performance of LLMs used as either direct EL executors or entity re-rankers underscore that LLMAEL’s context augmentation is the most effective way to integrate LLMs into EL.

5.4 Error Analysis

To gain deeper insight into how LLMAEL’s context augmentation affects EL performance, we conducted a manual error analysis of failure cases. Specifically, we focused on the WNED-WIKI dataset due to its intermediate size and comprehensive content derived from the Wikipedia encyclopedia.

Upon examining the test outputs of LLMAEL × ReFinED_{FT} on WNED-WIKI, we observed that LLMAEL context augmentation improved 2.67% of the test cases, turning them from incorrect to correct, demonstrating its usefulness in enhancing EL performance. However, 1.19% of the test cases regressed, turning from correct to incorrect, indicating that LLMAEL can also introduce errors in certain scenarios. To better understand these failure cases, we

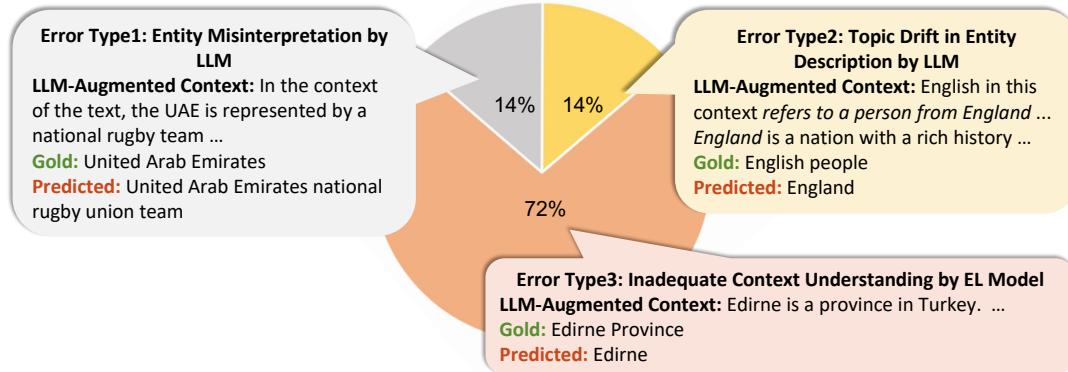


Figure 3: Error analysis of LLMAEL.

randomly sampled 50 entries from the 1.19% of regressed cases and performed a detailed manual error analysis.

Our results are illustrated in Figure 3, along with representative examples. The failure cases can be categorized into three types:

- (1) **Entity Misinterpretation by LLM (14%)**: In this category, the LLM incorrectly interprets the target entity and generates an inaccurate description. For example, as shown in Figure 3, while the original context refers to the UAE as the United Arab Emirates, the LLM misinterprets it as the UAE national rugby team, generating a description of the team. This misleads the EL model to link to the incorrect entity “United Arab Emirates national rugby union team.” We attribute this error type to the LLM’s misinterpretation during context generation.
- (2) **Topic Drift in LLM Entity Description (14%)**: Here, the LLM correctly identifies the target entity but generates context that diverges to describe another related entity. For example, as illustrated in Figure 3, the LLM correctly interprets the mention “English” as referring to English people but produces a lengthy description of the nation of England, which is a separate entity. This misleads the EL model to link to “England” instead of “English people.” This type of error stems from the LLM’s inability to maintain focus on the intended entity.
- (3) **Inadequate Context Understanding by EL Models (72%)**: This is the largest category of errors. In these cases, the LLM-generated context is entirely accurate, but the EL model still produces incorrect results. Such failures may arise from either (a) inherent limitations in the EL model’s capabilities, or (b) the EL model’s inability to fully adapt to the token distribution of the augmented context, even after fine-tuning. The latter suggests that while fine-tuning improves the compatibility of EL models with augmented contexts, there still remain cases where the models fail to adapt effectively.

Overall, the analysis shows that LLMs correctly interpret target entities in most cases, as indicated by the combined 86% share of error types 2 and 3, supporting their strength in entity interpretation and potential for EL tasks. Still, error type 1 reveals occasional misinterpretations, while type 2 shows that even accurate interpretations can be undermined by off-topic context. The high proportion of type 3 further highlights the challenge of fully adapting EL models to LLM-augmented contexts.

6 Conclusion

In this paper, we proposed LLMAEL, a novel framework that augments specialized EL models with LLM-generated context. We also explored alternative strategies for integrating LLMs into the EL process, including direct EL execution and entity re-ranking. Compared to these alternatives, LLMAEL’s context augmentation strategy provides the most effective use of LLMs for EL, achieving new state-of-the-art results on 5 of 6 benchmarks in the tuning-free setting and all 6 with lightweight EL model fine-tuning. We further validated LLMAEL’s robustness across three LLM backbones and explored ensemble techniques to further boost performance. Our analysis shows that LLM context augmentation consistently benefits mid- to long-tail entities, underscoring its strong potential for advancing entity linking.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62476150), Beijing Natural Science Foundation (L243006), and Tsinghua University (Department of Computer Science)-Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things (JCIOT). The authors also thank the Deng Feng Fund for supporting this research, and all anonymous reviewers and meta-reviewers for their valuable feedback.

GenAI Usage Disclosure

In this work, generative AI tools were utilized during the data construction phase. Specifically, given a target mention and its original context, we employed off-the-shelf large language models to generate additional contextual information to support entity linking. No generative AI tools were used in the writing of this paper, apart from basic spelling and grammar checks.

References

- [1] Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 209–220. doi:10.18653/v1/2022.naacl-industry.24
- [2] Salvador V Balkus and Donghui Yan. 2022. Improving short text classification with augmented data using GPT-3. *Natural Language Engineering* (2022), 1–30. <https://www.cambridge.org/core/journals/natural-language-engineering/article/improving-short-text-classification-with-augmented-data-using-gpt3/4F23066E3F0156382190BD76DA9A7BA5>
- [3] Giacomo Balloccu, Ludovico Boratto, Gianni Fenu, and Mirko Marras. 2022. Post processing recommender systems with knowledge graphs for recency, popularity, and diversity of explanations. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. <https://dl.acm.org/doi/pdf/10.1145/3477495.3532041>
- [4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfbcb4967418bfb8ac142f64a-Abstract.html>
- [5] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive Entity Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net. <https://openreview.net/forum?id=5k8F6U39V>
- [6] Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. 2023. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. *ArXiv preprint abs/2310.07289* (2023). <https://arxiv.org/abs/2310.07289>
- [7] Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, 708–716. <https://aclanthology.org/D07-1074>
- [8] Wen Cui, Leanne Rolston, Marilyn Walker, and Beth Ann Hockey. 2022. OpenEL: An Annotated Corpus for Entity Linking and Discourse in Open Domain Dialogue. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2245–2256. <https://aclanthology.org/2022.lrec-1.241>
- [9] Yifan Ding, Qingkai Zeng, and Tim Weninger. 2024. ChatEL: Entity Linking with Chatbots. *arXiv preprint arXiv:2402.14858* (2024).
- [10] Evgeniy Gabrilovich, Michael Ringgaard, Amarnag Subramanya, et al. 2013. FACC1: Freebase annotation of ClueWeb corpora. <http://lemurproject.org/clueweb09/FACC1/>
- [11] Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web* 9, 4 (2018), 459–479. <https://dl.acm.org/doi/abs/10.3233/SW-170273>
- [12] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK, 782–792. <https://aclanthology.org/D11-1072>
- [13] Phong Le and Ivan Titov. 2018. Improving Entity Linking by Modeling Latent Relations between Mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1595–1604. doi:10.18653/v1/P18-1148
- [14] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated Knowledge Prompting for Commonsense Reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 3154–3169. doi:10.18653/v1/2022.acl-long.225
- [15] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-Shot Entity Linking by Reading Entity Descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3449–3460. doi:10.18653/v1/P19-1335
- [16] David Milne and Ian H Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 509–518. <https://dl.acm.org/doi/abs/10.1145/1458082.1458150>
- [17] Yasumasa Onoe and Greg Durrett. 2020. Fine-Grained Entity Typing for Domain Independent Entity Linking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020*. AAAI Press, 8576–8583. <https://aaai.org/ojs/index.php/AAAI/article/view/6380>
- [18] Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yunjia Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, et al. 2023. When does In-context Learning Fall Short and Why? A Study on Specification-Heavy Tasks. *ArXiv preprint abs/2311.08993* (2023). <https://arxiv.org/abs/2311.08993>
- [19] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 1375–1384. <https://aclanthology.org/P11-1138>
- [20] Ulrich Rückert, Srinivas Sunkara, Abhinav Rastogi, Sushant Prakash, and Pranav Khaitan. 2022. A Unified Approach to Entity-Centric Context Tracking in Social Conversations. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 1275–1285. <https://aclanthology.org/2022.lrec-1.136>
- [21] Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2022. Neural entity linking: A survey of models based on deep learning. *Semantic Web* 13, 3 (2022), 527–570. <https://content.iospress.com/articles/semantic-web/sw222986>
- [22] Senbao Shi, Zhenran Xu, Baotian Hu, and Min Zhang. 2023. Generative multi-modal entity linking. *arXiv preprint arXiv:2306.12725* (2023).
- [23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint abs/2307.09288* (2023). <https://arxiv.org/abs/2307.09288>
- [24] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics* 9 (2021), 176–194. doi:10.1162/tacl_a_00360
- [25] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1929–1937. <https://dl.acm.org/doi/abs/10.1145/3534678.3539382>
- [26] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6397–6407. doi:10.18653/v1/2020.emnlp-main.519
- [27] Zilin Xiao, Ming Gong, Jie Wu, Xingyao Zhang, Linjun Shou, Jian Pei, and Daxin Jiang. 2023. Instructed language models with retrievers are powerful entity linkers. *arXiv preprint arXiv:2311.03250* (2023).
- [28] Amy Xin, Jinxin Liu, Zijun Yao, Zhicheng Lee, Shulin Cao, Lei Hou, and Juanzi Li. 2025. Atomr: Atomic operator-empowered large language models for heterogeneous knowledge reasoning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3344–3355.
- [29] Silei Xu, Shicheng Liu, Theo Culhane, Elizaveta Pertseva, Meng-Hsi Wu, Sina Semnani, and Monica Lam. 2023. Fine-tuned LLMs Know More, Hallucinate Less with Few-Shot Sequence-to-Sequence Semantic Parsing over Wikidata. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. <https://aclanthology.org/2023.emnlp-main.353>
- [30] Zijun Yao, Yantao Liu, Xin Lv, Shulin Cao, Jifan Yu, Juanzi Li, and Lei Hou. 2023. KoRC: Knowledge Oriented Reading Comprehension Benchmark for Deep Text Understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*. <https://aclanthology.org/2023.findings-acl.743>
- [31] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *ArXiv preprint abs/2209.10063* (2022). <https://arxiv.org/abs/2209.10063>