OneNet: A Fine-Tuning Free Framework for Few-Shot Entity Linking via Large Language Model Prompting

Anonymous ACL submission

Abstract

Entity Linking (EL) is the process of associating ambiguous textual mentions to specific entities in a knowledge base. Traditional EL methods heavily rely on large datasets to enhance their performance, a dependency that becomes problematic in the context of few-shot entity linking, where only a limited number of examples are available for training. To address this challenge, we present OneNet, an innovative framework that utilizes the few-shot learning capabilities of Large Language Models (LLMs) without the need for fine-tuning. To the best of our knowledge, this marks a pioneering approach to applying LLMs to few-shot entity linking tasks. OneNet is structured around three key components prompted by LLMs: (1) an entity reduction processor that simplifies inputs by summarizing and filtering out irrelevant entities, (2) a dual-perspective entity linker that combines contextual cues and prior knowledge for precise entity linking, and (3) an entity consensus judger that employs a unique consistency algorithm to alleviate the hallucination in the entity linking reasoning. Comprehensive evaluations across six benchmark datasets reveal that OneNet outperforms current state-of-the-art entity linking methods.

1 Introduction

Entity Linking (EL), also known as Named Entity Disambiguation (NED), entails the process of linking ambiguous textual mentions to specific entities in a knowledge base, as shown in Figure 1 (a). This process is a critical element of both Natural Language Processing (NLP) and Information Retrieval (IR) (Sevgili et al., 2022).

To enhance the accuracy of EL, researchers employ two primary methods: discriminative models and generative models. Discriminative models represent mentions and entities through embeddings and link entities by calculating the similarity. To augment the quality of embeddings, an expansive array of external features are adopted, including categorization of entities (Tedeschi et al., 2021), imposition of hierarchical constraints (Wang et al., 2023), and incorporation of pre-existing hyperlinks prior (Ayoola et al., 2022). Conversely, generative models inherently produce the linked entities by adjusting pre-trained language models through fine-tuning processes. These approaches formulate entity linking as various generative tasks, such as sequence-to-sequence constrained generation (De Cao et al., 2021), information extraction (Barba et al., 2022), question answering (Zhang et al., 2021), and instruction tuning (Xiao et al., 2023), to refine the performance.

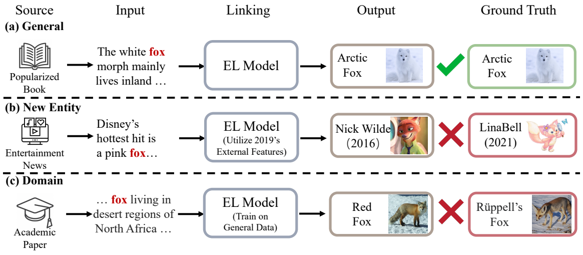

Source
Input
Linking
Output
Ground Truth
(a) General
Popularized Book
The white fox morph mainly lives inland …
EL Model
Arctic Fox
Arctic Fox
(b) New Entity
Entertainment News
Disney's hottest hit is a pink fox…
EL Model (Utilize 2019's External Features)
Nick Wilde (2016)
LinaBell (2021)
(c) Domain
Academic Paper
… fox living in desert regions of North Africa …
EL Model (Train on General Data)
Red Fox
Rüppell's Fox

Figure 1: An example of entity linking across various scenarios, where **mention** is bolded in red.

Despite promising results shown by conventional approaches, their pronounced reliance on extensive datasets limits their applicability in few-shot scenarios, where only limited annotated examples are available (Xu et al., 2023). This limitation manifests in two primary ways: Firstly, these methods depend heavily on predefined external features, which compromise their ability to accurately identify novel entities. As depicted in Figure 1 (b), the reliance on characteristics gathered in 2019 may lead to incorrect linking, such as misidentifying the 2021 Disney fox character, *Linnaeus Bell*, as the character *Nick Wilde* from 2016 due to outdated external features. Secondly, the dependency on large-scale training datasets poses significant challenges when adapting these models to specialized domains. Figure 1 (c) demonstrates that limited domain-specific data causes the model to be biased toward general data, potentially causing linking rare species like *Rüppell's fox* to more common ones like *Red fox*.

To mitigate the limitations inherent in conventional models that heavily rely on data, this study explores the utility of Large Language Models (LLMs) to enhance entity linking under a few-shot learning framework. The rationale for employing LLMs is grounded in their significant benefits: Firstly, LLMs excel at in-context learning (Dong et al., 2022), which allows them to reason about previously unencountered questions using a minimal set of examples. Second, during pre-training on rich datasets (e.g., Wikipedia), LLMs have acquired versatile prior knowledge, ensuring their proficiency across a multitude of specialized fields (Touvron et al., 2023).

However, the utilization of LLMs for entity linking encounters several notable challenges. **1) Token Length Limitations**. Entity linking typically recalls a substantial volume of candidate entities and their corresponding descriptions, often exceeding 10,000 tokens, which surpasses the capacity of most LLMs. **2) Reasoning Balance.** Enhanced reasoning techniques such as CoT (Wei et al., 2022b) could inadvertently suppress a model's inherent knowledge (Wei et al., 2023). Maintaining a balance between analytical inference and prior knowledge is crucial for effective EL. **3) Hallucination.** Hallucination in LLM reasoning emerges as a critical obstacle, especially for complex reasoning tasks such as EL. (Ji et al., 2023). Identifying and rectifying such reasoning errors in entity linking poses a persistent challenge.

To tackle these obstacles, we introduce **OneNet**, a comprehensive framework composed of various interconnected modules, each prompted by the LLMs. To our knowledge, in few-shot entity linking, this constitutes the inaugural effort to apply LLMs without fine-tuning. Specifically, our approach first begins with the innovative Entity Reduction Processor (ERP), which is designed to condense the input text by summarizing entity descriptions and filtering irrelevant entities. Second, to maintain an equilibrium in the analytical process, we introduce the Dual-perspective Entity Linker (DEL), which executes EL by integrating contextual cues with prior knowledge. Third, we address the hallucination problem in EL through our Entity Consensus Judger (ECJ). It undertakes a comparative analysis of two results from DEL, and further employs a consistency algorithm to rectify errors in the reasoning process of LLMs. Finally, the efficacy of OneNet is underscored across six diverse datasets, which demonstrate the superiority of our proposed method. All prompts are shown in Appendix C, while anonymous codes are available at https://anonymous.4open.science/r/LLM4EL/.

2 Related Works

2.1 Entity Linking

Entity linking (EL) is integral to information extraction and natural language processing. Initial studies focused on discriminant models, employing external datasets for enhanced entity representation. Techniques included leveraging mention-entity hyperlink counts in Deep-ed (Ganea and Hofmann, 2017) and Mulrel-nel (Le and Titov, 2018), augmenting entity embeddings with NER classifiers in NER4EL (Tedeschi et al., 2021), and introducing hierarchical constraints in CDHCN (Wang et al., 2023). Other approaches enhanced their models by leveraging vast quantities of data (e.g., Wikipedia, Wikidata). Blink (Wu et al., 2020) trained bi-encoders on 5.9 million entities to incorporate entity titles and descriptions. EntQA (Zhang et al., 2021) further refined their methods using question-answering techniques and data from Blink, while ReFinED (Ayoola et al., 2022) fused priors, types, and descriptions using over 6 million entities with their hyperlinks. Nonetheless, the reliance on large-scale data limits their application to the new or domain-specific entities.

Recently, generative models for entity linking have witnessed a surge in research. Genre (De Cao et al., 2021) generated the predicted entities directly after the mention by a constrained decoder. Extend (Barba et al., 2022) then directly extracted the linking entities from the context with candidate entities. Meanwhile, InsGen (Xiao et al., 2023) proposed an instruction-tuning method on LLM to generate results. Unfortunately, these methods still rely on fine-tuning to inject prior knowledge into the models, limiting their adaptability to adapt to different scenarios with a few examples.

Notably, previous studies (Wu et al., 2020; Ayoola et al., 2022) claimed the suitability of their methods for few-shot and zero-shot learning, yet primarily in out-of-domain contexts. In contrast, our study focuses on a more realistic few-shot

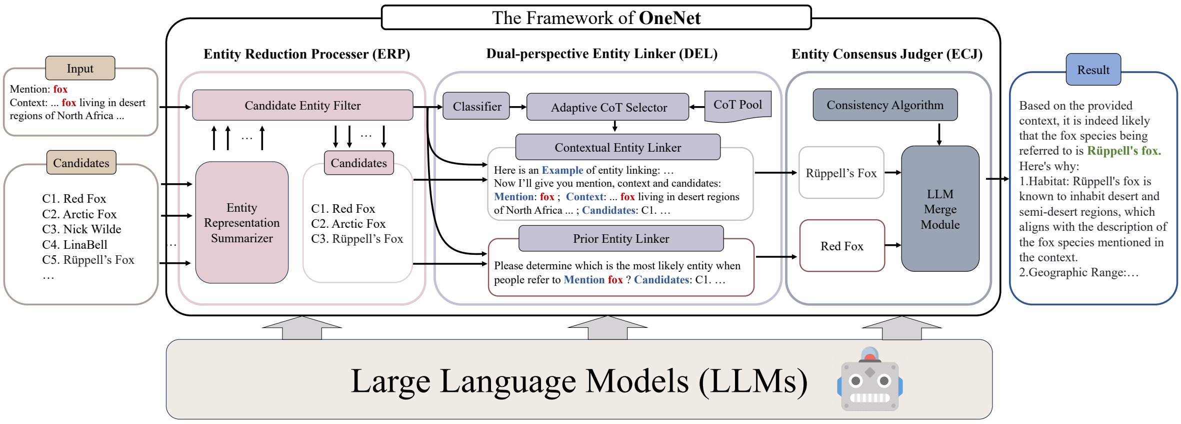

The Framework of OneNet
Entity Reduction Processer (ERP)
Dual-perspective Entity Linker (DEL)
Entity Consensus Judger (ECJ)
Input
Mention: fox
Context: ... fox living in desert regions of North Africa ...
Candidates
C1. Red Fox
C2. Arctic Fox
C3. Nick Wilde
C4. LinaBell
C5. Rüppell's Fox
…
Candidate Entity Filter
Entity Representation Summarizer
Candidates
C1. Red Fox
C2. Arctic Fox
C3. Rüppell's Fox
Classifier
Adaptive CoT Selector
CoT Pool
Contextual Entity Linker
Here is an Example of entity linking: …
Now I'll give you mention, context and candidates:
Mention: fox ; Context: ... fox living in desert regions of North Africa ... ; Candidates: C1. …
Prior Entity Linker
Please determine which is the most likely entity when people refer to Mention fox ? Candidates: C1. …
Consistency Algorithm
Rüppell's Fox
Red Fox
LLM Merge Module
Result
Based on the provided context, it is indeed likely that the fox species being referred to is Rüppell's fox.
Here's why:
1.Habitat: Rüppell's fox is known to inhabit desert and semi-desert regions, which aligns with the description of the fox species mentioned in the context.
2.Geographic Range:…
Large Language Models (LLMs)

Figure 2: The illustration of OneNet framework, which contains three distinct modules: (a) Entity Reduction Processor (ERP), (b) Dual-perspective Entity Linker (DEL), and (c) Entity Consensus Judger (ECJ).

framework (Xu et al., 2023). To the extent of our knowledge, this study is the inaugural exploration of leveraging LLMs for the few-shot entity linking without any necessity for model fine-tuning.

2.2 Large Language Models

Large language models (LLMs), including GPT (Du et al., 2022), Llama (Touvron et al., 2023), and GLM (Du et al., 2022), demonstrate impressive few-shot learning ability in numerous natural language processing (NLP) tasks (Liu et al., 2023a). This emergent capacity allows them to outperform earlier supervised approaches and even achieve human-level performance on certain tasks, all without fine-tuning (Wei et al., 2022a).

However, applying LLMs to complex problem-solving remains challenging (Feng et al., 2023). One way to improve LLM's reasoning is Chain-of-Thought (CoT) (Wei et al., 2022b), which has attracted growing interest. Some research explored optimizing example selection based on similarity (Rubin et al., 2022), diversity (Zhang et al., 2022), and complexity (Fu et al., 2022). Other efforts were directed at designing effective reasoning pipelines. For instance, Least2Most (Zhou et al., 2022) suggested simplifying complex problems into manageable subproblems. SICoT (Creswell et al., 2022) proposed a Selection-Inference framework. Furthermore, Deductive CoT (Ling et al., 2023) addressed hallucination issues through a sequential reasoning verification process. Despite these advancements, the application of LLMs in EL necessitates additional investigation.

3 Preliminary

3.1 Few-shot Entity Linking

In this paper, we formally define m as a mention in a text $\mathcal{S}$, and e as an entity in a knowledge base (KB) associated with its description. For each mention m, we have a pre-processing step called entity candidate generation that chooses potential candidate entities $\theta = \{e_1, e_2, ..., e_n\}$ from a specific KB. Each mention m also has a labeled link entity y. Following the few-shot setting (Xu et al., 2023), the training set $D_{train} = \{(S, m, \theta, y)\}$ contains only a few data and satisfies $|D_{train}| \ll |D_{test}|$. Our goal is to learn the input (S, m, θ) to output y mapping with as little training data as possible.

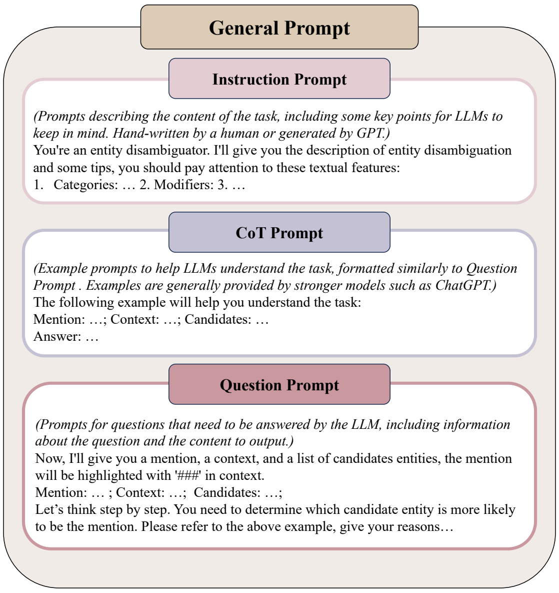
3.2 Entity Linking with LLMs

As illustrated in Figure 3, we form a query for LLMs as $q = [m; S; \theta]$, and the prompt P of entity linking can be composed as a task-specific instruction I, n CoT exemplars and the test query itself:

$$P = [I; q_1; y_1, ..., q_n; y_n; q_{test}], \qquad (1)$$

where $y = (e, r)$ is the output of LLMs, which contains a predicted entity e and the reasoning r.

It is important to acknowledge that the difficulties outlined in Section 1 present substantial impediments to the entity linking process when a single Large Language Model (LLM) is employed. Therefore, we form a pipeline to complete the entity linking by adjusting instruction I and exemplars to prompt multiple modules with different functions. Specific prompts can be found in the Appendix C.



(Prompts describing the content of the task, including some key points for LLMs to keep in mind. Hand-written by a human or generated by GPT.)
You're an entity disambiguator. I'll give you the description of entity disambiguation and some tips, you should pay attention to these textual features:
1. Categories: … 2. Modifiers: 3. …

(Example prompts to help LLMs understand the task, formatted similarly to Question Prompt . Examples are generally provided by stronger models such as ChatGPT.)
The following example will help you understand the task:
Mention: …; Context: …; Candidates: …
Answer: …

(Prompts for questions that need to be answered by the LLM, including information about the question and the content to output.)
Now, I'll give you a mention, a context, and a list of candidates entities, the mention will be highlighted with '###' in context.
Mention: … ; Context: …; Candidates: …;
Let’s think step by step. You need to determine which candidate entity is more likely to be the mention. Please refer to the above example, give your reasons…

Figure 3: The Illustration of General Prompt Structure

4 Method

4.1 An Overview of One-Net

As shown in Figure 2, the proposed methodology comprises three distinct modules: (a) Entity Reduction Processor (ERP), (b) Dual-perspective Entity Linker (DEL), and (c) Entity Consensus Judger (ECJ). Initially, ERP conducts a two-step process that involves the summarizing of entity descriptions and the point-wise exclusion of irrelevant candidate entities. Subsequently, DEL is devised to establish fine-grained entity linking within the filtered candidates, utilizing both contextual analysis and prior knowledge. Finally, ECJ combines the contextual result and prior result to generate the predicted entity. Notably, each module is derived from a large language model, leveraging distinct prompts without fine-tuning.

4.2 Entity Reduction Processor

To address the issue of token length limitations in Section 1, we employed the following optimization strategy. As illustrated in Figure 2, firstly, the Entity Representation Aggregator is utilized to condense the descriptions of entities, thereby providing a more succinct representation. Secondly, the Candidate Entity Filter is implemented to execute an initial, point-wise filtration of potential entities to reduce the number of candidates.

4.2.1 Entity Representation Summarizer

From previous research (Cheng et al., 2015), entity summarization can significantly enhance the efficiency of entity linking by distilling the essential characteristics of entity descriptions. In this context, we function as a summarizer by engaging a large language model through a simple prompt. The prompt P_{sum} consists solely of a summary instruction prompt and an entity, which are structured as follows:

$$P_{sum} = [I_{sum}; e], \tag{2}$$

where I_{sum} is the summary instruction prompt, e is an entity with its description.

4.2.2 Candidate Entity Filter

In light of the suboptimal performance exhibited by directly list-wise EL, the Candidate Entity Filter transfers the list-wise EL into a sequence of point-wise EL, which only has one candidate in the query. This strategic conversion facilitates the effective filtration of irrelevant entities, which prompt P_{fil} is as follows:

$$P_{fil} = [I_{el}; m; S; e_i], \tag{3}$$

where I_{el} is the instruction of entity linking, e_i is one entity in the candidates. To improve efficiency, we don’t use the Chain-of-Thought methods. Inspired by the insights of prior research (Honovich et al., 2022), we utilize LLMs to formulate instructions, which details are shown in Appendix C.

4.3 Dual-perspective Entity Linker

Building upon the established understanding from prior research (Ganea and Hofmann, 2017; Le and Titov, 2018), it is recognized that entity linking can be decomposed into two distinct components: a prior probability and a contextual probability, which can be succinctly expressed through the following formalization:

$$P(e|c) \propto P(e)P(c|e), \tag{4}$$

where $P(e)$ is the link probability without context, $P(c|e)$ can be seen as the relevance of the entity and the context.

Accordingly, to maintain an equilibrium in the analytical process mentioned in Section 1, the Dual-perspective Entity Linker is composed of two components in Figure 2: the Contextual Entity Linker, which leverages the inferential capabilities of LLMs to generate context-aware predictions,

Algorithm 1 The Consistency Algorithm

```
Input: contextual prediction $e_{context}$, prior prediction $e_{prior}$, mention $m$, context $S$
Output: link entity $e_{result}$
if $e_{context} = e_{prior}$ then
    $e_{result} \leftarrow e_{context}$
else
    $e_{result} \leftarrow LLM(e_{context}, e_{prior}, m, S)$
end if
return $e_{result}$
```

and the Prior Entity Linker, which harnesses the inherent knowledge embedded within LLMs to produce predictions based on prior information.

4.3.1 Contextual Entity Linker

To effectively harness the inferential capabilities of LLMs for list-wise entity linking, the Contextual Entity Linker employs a structured prompt as depicted in Figure 3. This prompt is composed of three distinct segments: the instruction prompt, the CoT prompts, and the question prompt, which is formed as:

$$P_{context} = [I_{el}; q_1; y_1, ..., q_n; y_n; q_{test}], \quad (5)$$

where I_{el} is the entity linking instruction mentioned in Section 4.2.2, $[q_i, y_i]$ is the CoT exemplar.

CoT Exemplar Pool. Inspired by previous work (Liu et al., 2023b), in order to distill the reasoning power of the advanced models, we sample a subset of questions from the training dataset and present them to advanced models for response. To mitigate the issue of hallucination(Ji et al., 2023), we implement a stringent selection criterion, retaining only those responses that accurately predict the correct entity.

Adaptive CoT Selector. To effectively determine the optimal CoT reasoning approach, our selection process is informed by two critical dimensions: context similarity and entity category. Firstly, we postulate that similar contexts likely share analogous reasoning patterns. To implement this, we quantify the resemblance by computing the cosine similarity between the input context and the exemplar contexts in our CoT pool. Secondly, we recognize that mentions belonging to the same category often exhibit common features that are pertinent to the reasoning process. To leverage this, we employ an LLM as a classifier, which incorporates a specific classifier instruction prompt along with the mention and its provided context. Ultimately, our composite CoT score is derived by integrating these considerations:

$$s = \alpha \cdot cos(S_i, S_{test}) + (1-\alpha) \cdot \mathbb{I}(m_i, m_{test}), \quad (6)$$

where $\mathbb{I}(\cdot, \cdot)$ indicates whether the category is the same in both mentions and α is a hyperparameter.

4.3.2 Prior Entity Linker

To utilize the inherent prior knowledge in the LLMs, we employ an LLM as the Prior Entity Linker. As shown in figure 2, the prior prompt is comprised of three distinct components: the prior instruction prompt, the mention, and the filtered candidates, which can be represented as follows:

$$P_{prior} = [I_{prior}; m; \theta_{fil}]. \quad (7)$$

It is worth noting that the context is hidden to prevent the influence of noise in the context on the prior (Conover et al., 2018). Due to the lack of context, many of the hints about the context in the entity linking instructions are no longer appropriate, so we use instructions that are not the same as the contextual linker. Additionally, the imperative for preserving prior knowledge necessitates the exclusion of CoT methods to preclude the potential overwriting of LLM intrinsic knowledge (Wei et al., 2023).

4.4 Entity Consensus Judger

To ensure accurate entity prediction from the two predicted entities in Section 4.3, the Entity Consensus Judger utilizes a consistency algorithm to mitigate potential hallucination in DEL, as illustrated in Figure 2. The algorithm functions as follows: when both prediction modules concur on the same entity, that entity is confirmed as the result. Conversely, in instances of prediction discordance, the ECJ invokes an auxiliary LLM to ascertain the correct entity for linking. The details of this algorithm are shown in Algorithm 1.

The propensity for inaccuracies within the Contextual Linker predominantly stems from misleading of CoT. Conversely, errors within the Prior Linker are principally attributed to the lack of context. To mitigate the occurrence of both error types, the auxiliary LLM has been designed to incorporate instruction prompt, context, and the entities ascertained by the dual linkers, which is formed as:

$$P_{merge} = [I_{el}; m; S; e_{context}; e_{prior}], \quad (8)$$

where I_{el} is the entity linking prompts as Section 4.2.2. Candidate entities are limited to the entities predicted by the previous linkers.

Dateset	ACE2004	AIDA	AQUAINT	CWEB	MSNBC	WIKI
Mentions	257	4463	727	11154	656	6821

Table 1: The Statistics of Test Datasets

5 Experiments

5.1 Datasets

For the reliability and authority of experimental results, we have conducted evaluations across six widely recognized datasets: ACE2004, AIDA(Hoffart et al., 2011), AQUAINT(Guo and Barbosa, 2018), CWEB, MSNBC, and WIKI (Evgeniy et al., 2013). Table 1 provides further information about the datasets. We utilize the November 2020 snapshot of English Wikipedia (Tedeschi et al., 2021) as our knowledge base (KB). Following the previous work (Wang et al., 2023), we employed an alias table to generate the candidate entities. For efficiency, we limited the candidate pool to 10 entities for datasets with large numbers of mentions, such as AIDA, CWEB, and WIKI. For the remaining datasets, we retained the complete set of candidate entities.

5.2 Implementation Details

We implement our method on Zephyr-7b-beta (Tunstall et al., 2023). The exemplar pool, comprising 65 data instances, is derived from the training set of AIDA. We place $n = 1$ exemplar in the prompt P for the contextual entity linker. The adaptive CoT selector's hyper-parameter is set to $\alpha = 0.5$. When running Zephyr, we fix the parameters to the default values provided by the official, and the max new token is set to 1024. For classifier, we use Wikipedia's 12 categories. To mitigate the potential bias arising from sequence dependency within the model, we randomize the order of candidate entities for each time. We take the first occurrence of the entity as the prediction. Following the previous work (Sevgili et al., 2022), we report the micro F1 to assess entity linking performance.

5.3 Benchmark Methods

To evaluate the effectiveness of One-Net, we compare it with traditional state-of-the-art supervised methods and popular large language models:

- **Traditional Supervised Methods.** These models necessitate supervised learning. Specifically, Mulrel-nel (Le and Titov, 2018), NER4EL (Tedeschi et al., 2021), and CDHCN (Wang et al., 2023) utilize leverage external data to train discriminative models. Conversely, Extend (Barba et al., 2022) employs a generative approach to extract the corresponding entity from the candidate entities. Our exemplar pool provides the foundational data required for the training.

- **Large Language Models.** Since entity linking is a text-only task, LLMs can also be directly applied to it. For GLM (Du et al., 2022), We tested both 4K and 32K versions to confirm the effectiveness of long text training. For Zephyr (Tunstall et al., 2023), we use the same version to validate the effectiveness of our framework. For ChatGPT (Brown et al., 2020), we utilize the model *gpt-3.5-turbo-1106* to test. During the generation of outputs, we adhered to the default settings provided by the official documentation. The same exemplars are provided to all LLMs to facilitate their chain-of-thought ability.

As we focus on the few-shot scenario, we disregard additional models which are trained on massive additional data, such as Blink (Wu et al., 2020), EntQA (Zhang et al., 2021), and ReFinED (Ayoola et al., 2022), to ensure an equitable comparison. Additionally, we have omitted results from other popular language models like Llama2 (Touvron et al., 2023), as their performance is found to be suboptimal, which falls below 5%.

5.4 Experimental Results

The results of all methods on the datasets are shown in Table 2. We report three OneNet results based on different base models. In general, OneNet with Zephyr has achieved the best performance compared with these SOTA baselines. Specifically, our method outperforms the best-performing baseline (i.e., Extend, ChatGPT) by 4%-11%. Additionally, we discover some interesting phenomena:

First, traditional generative models, such as Extend, demonstrate superior performance over traditional discriminative models like NER4EL and CDHCN in few-shot scenarios, supporting the idea of generative LLMs for few-shot entity linking. Second, a single LLM does not have good entity linking capability. For instance, all of OneNet's results are significantly better than the corresponding single model's results (i.e., GLM, Zephyr). These findings underscore the necessity of using multi-

Dateset		ACE2004	AIDA	AQUAINT	CWEB	MSNBC	WIKI
Tradition	Mulrel-nel	0.217	0.328	0.262	0.267	0.422	0.380
	NER4EL	0.531	0.569	0.460	0.488	0.602	0.495
	Extend	0.604	0.563	0.641	0.537	0.715	0.506
	CDHCN	0.438	0.575	0.465	0.504	0.654	0.505
LLMs	GLM-8K[1]	0.482	0.520	0.466	0.454	0.550	0.550
	GLM-32K	0.447	0.439	0.431	0.487	0.584	0.532
	Zephyr	0.467	0.322	0.495	0.518	0.637	0.555
	ChatGPT	0.611	0.451	0.560	0.546	0.732	0.615
OneNet	GLM-8K	0.611	0.639	0.626	0.587	0.713	0.626
	GLM-32K	0.650	0.672	0.626	0.606	0.764	0.645
	Zephyr	**0.681**	**0.690**	**0.686**	**0.650**	**0.796**	**0.676**

[1] The data used for GLM-8K is filtered by our Entity Reduction Processor.

Table 2: Micro-F1 Scores of Few-shot Entity Linking on Various Datasets

Dateset	ACE2004	AIDA	AQUAINT	CWEB	MSNBC	WIKI
Recall	0.765	0.885	0.844	0.802	0.880	0.875
Filtering Rate	0.900	0.648	0.865	0.643	0.825	0.622
Avg	4.07	2.55	4.72	2.51	4.57	2.31

Table 3: Recall, Filtering Rate, and Average Candidates across Datasets Filtered by ERP.

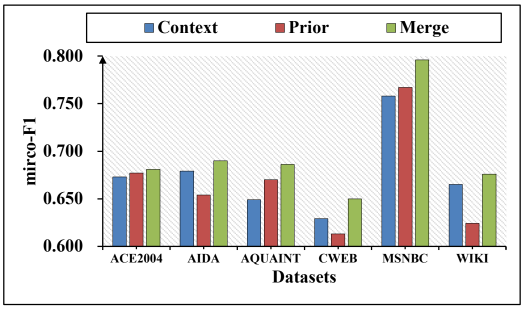


Figure 4: Comparison of Prior, Context, and Merge

ple prompts to leverage the diverse capabilities of LLMs for effective entity linking.

5.5 Ablation Study

5.5.1 Accuracy and Efficiency of ERP

To validate the effectiveness of the ERP in Section 4.2, we show the recall, filtering rate, and average number of remaining candidate entities on all datasets, where filter rating shows the percentage of filtered-out candidates. As shown in Table 3, recall reaches 0.8 for most datasets except ACE2004, and for filtering rate and Avg, the filtering rate reaches more than 0.8 on the unprocessed dataset, and the Avg is around 4. For the preprocessed dataset, the filtering rate reaches more than 0.6 and the Avg is around 3. All these prove that ERP can filter out as many irrelevant entities as possible while ensuring that the correct entities are retained.

5.5.2 Context and Prior are Both Necessary

As we mentioned in Section 4.3, to substantiate the indispensability of both context and prior perspectives, a comparative analysis of the individual modules and their merged results is conducted. Figure 4 illustrates that the ECJ module yields superior performance across all datasets, thereby affirming its efficacy. Nonetheless, it is noteworthy that the context and prior components each exhibit distinct advantages across varying datasets. For example, context outperforms 2% on AIDA, while prior is 2% higher on AQUAINT. This observation validates the rationale behind incorporating dual perspectives within the DEL module.

5.5.3 Detailed Instructions and Reasonable Exemplars make LLMs Aligned

Figure 3 elucidates that our prompt contains both detailed instructions and reasonable exemplars to facilitate LLM's understanding of entity linking. The comparative results, as summarized in Figure 6, demonstrate that our adaptive CoT approach surpasses other CoT selection methods across all evaluated datasets, which underscores the efficacy of our method in identifying more suitable exemplars. Meanwhile, our findings indicate that the absence of detailed instructions hampers the LLM's ability to understand the EL task (e.g., No-Prompt, W/o Ins). Furthermore, our analysis reveals that prompts with CoT demonstrate superior performance in ACE2004 and AIDA. Conversely, prompts without CoT exhibit enhanced efficacy in AQUAINT. This result is consistent with Figure 4, as AIDA is more context-aware, AQUAINT is more prior-dependent and ACE2004 considers both. This further suggests that both reasoning

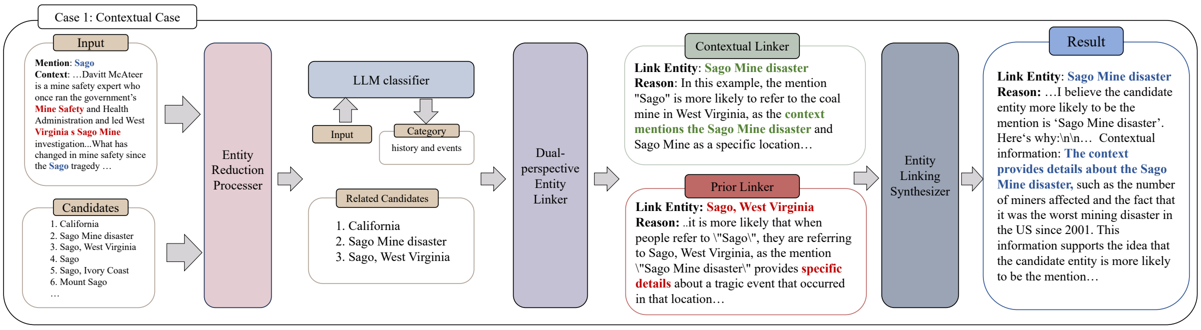

Case 1: Contextual Case
Input
Mention: Sago
Context: …Davitt McAteer is a mine safety expert who once ran the government's Mine Safety and Health Administration and led West Virginia s Sago Mine investigation...What has changed in mine safety since the Sago tragedy …
Candidates
1. California
2. Sago Mine disaster
3. Sago, West Virginia
4. Sago
5. Sago, Ivory Coast
6. Mount Sago
…
Entity Reduction Processer
LLM classifier
Input
Category
history and events
Related Candidates
1. California
2. Sago Mine disaster
3. Sago, West Virginia
Dual-perspective Entity Linker
Contextual Linker
Link Entity: Sago Mine disaster
Reason: In this example, the mention "Sago" is more likely to refer to the coal mine in West Virginia, as the context mentions the Sago Mine disaster and Sago Mine as a specific location…
Prior Linker
Link Entity: Sago, West Virginia
Reason: ..it is more likely that when people refer to \"Sago\", they are referring to Sago, West Virginia, as the mention \"Sago Mine disaster\" provides specific details about a tragic event that occurred in that location…
Entity Linking Synthesizer
Result
Link Entity: Sago Mine disaster
Reason: …I believe the candidate entity more likely to be the mention is 'Sago Mine disaster'. Here's why:\n\n… Contextual information: The context provides details about the Sago Mine disaster, such as the number of miners affected and the fact that it was the worst mining disaster in the US since 2001. This information supports the idea that the candidate entity is more likely to be the mention…

Figure 5: Case study of OneNet. Key information in context makes the contextual linker reason correctly.

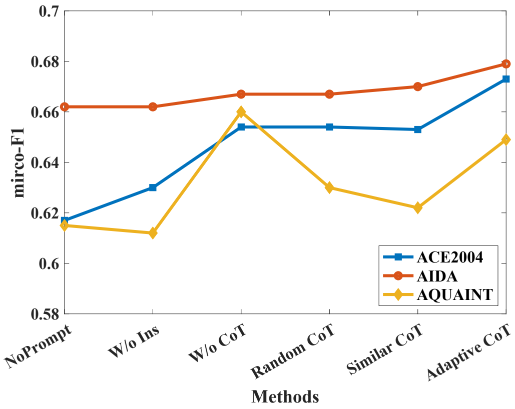


Figure 6: The Impact of Different Prompt Structure

ability and prior knowledge are important for EL.

To test the robustness of our module, we generate various instruction prompts for testing. The results, illustrated in Figure 7 in Appendix A, demonstrate that the module's performance remained stable despite variations in the prompts. Additionally, Table 5 in Appendix A presents an example of repeated answers, emphasizing that the semantic of outputs is invariant within the framework constraints. These consistency underscores the user-friendliness of our module, demonstrating its ability to perform reliably under diverse instructions.

5.6 Case Study of OneNet

For a more intuitive comparison of how our frameworks work, we provide two case studies, one utilizing contextual linking in Figure 5 while the other utilizing prior linking in Figure 8 in Appendix A.3.

The first mention is *Sago*, found in an article on mining safety. Initially, the Entity Reduction Processor screened out 6 irrelevant entities. For instance, *Sago* as a foodstuff and *Mount Soga* for its geographical inaccuracy. Subsequently, three pertinent entities remained: *Sago Mine disaster*, *Sago, West Virginia*, and *California*. Following the classification *history and events*, the contextual linker identified *Sago Mine disaster* as the likely reference, deducing that *Sago Mine* was implied within the text. Conversely, the prior linker suggested *Sago, West Virginia*, which considers *Sago Mine disaster* to be overly specific. Ultimately, the Entity Consensus Judger favored the contextual prediction *Sago Mine disaster*, corroborated by the text's detailed description of the event. This resolved an error in the prior linker by taking into account the context provided.

The second entity mentioned is *Orange County* in an airport blog. The procedure mirrors that of the initial case. However, the term "airport" in the context notably causes the contextual linker's error. In contrast, the prior linker predominantly depends on the model's intrinsic knowledge to render an accurate prediction. Details are provided in Figure 8 in Appendix A.3.

More experimental analyses, such as Framework Generalization, can be found in Appendix B.

6 Conclusion

In this study, we introduced OneNet, a novel framework for few-shot entity linking by leveraging large language model prompts without fine-tuning. Specifically, OneNet was comprised of three key LLM-prompted components: the Entity Reduction Processor, which was designed for efficient text condensation by summarizing entity descriptions and irrelevant entity filtering; the Dual-perspective Entity Linker, which considered both contextual information and prior knowledge to provide a balanced analysis; and the Entity Consensus Judger, which was instrumental in reducing hallucinations through a consistency merger algorithm. Our framework demonstrated superior performance on six datasets. Our future research will aim to merge mention detection within our model.

7 Limitations

Although we have demonstrated the superiority of our OneNet compared to previous work on six real-world datasets, there are still two limitations that should be addressed in the future:

(1) Since our framework is based on prompting LLMs, the efficiency of the whole pipeline operation will be limited by the inference speed of LLMs. As shown in the Table 4 in Appendix A, the runtime of our framework is heavily influenced by the base model. Nonetheless, the field has witnessed significant advancements aimed at expediting the inference process for LLMs. These enhancements encompass strategies like I/O optimization (Dao et al., 2022), model pruning (Liu et al., 2023c), and quantization techniques (Dettmers et al., 2022),. It is our assertion that these ongoing research efforts will eventually surmount the current limitations imposed by the inference speed of large language models, thereby mitigating this bottleneck in the foreseeable future.

(2) Currently, our framework is dedicated exclusively to the task of entity disambiguation. It is important to note that the broader domain of entity linking encompasses both entity disambiguation and mention detection. Actually, mention detection has been effectively approached using large language models (Jin et al., 2023) and prompting techniques (Shen et al., 2023), its integration is not only complementary but can also enhance the performance of entity disambiguation. In future work, we will explore more efficient ways to integrate entity disambiguation and mention detection.

References

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*, pages 209–220.

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2022. Extend: Extractive entity disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2478–2488.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Gong Cheng, Danyun Xu, and Yuzhong Qu. 2015. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *Proceedings of the 24th International Conference on World Wide Web*, pages 184–194.

Michael Conover, Matthew Hayes, Scott Blackburn, Pete Skomoroch, and Sam Shah. 2018. Pangloss: Fast entity linking in noisy text environments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 168–176, New York, NY, USA. Association for Computing Machinery.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

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.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. 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.

Gabrilovich Evgeniy, Ringgaard Michael, and Subramanya Amarnag. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0).

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4):459–479.

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*, pages 782–792.

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2022. Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Xiaomeng Jin, Bhanukiran Vinzamuri, Sriram Venkatapathy, Heng Ji, and Pradeep Natarajan. 2023. Adversarial robustness for large language NER models using disentanglement and word attributions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12437–12450, Singapore. Association for Computational Linguistics.

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)*, pages 1595–1604.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023b. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4549–4560, New York, NY, USA. Association for Computing Machinery.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, et al. 2023c. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning*, pages 22137–22176. PMLR.

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*, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. 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.

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):527–570.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt locating and typing for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12492–12507, Toronto, Canada. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Simone Tedeschi, Simone Conia, Francesco Cecconi, and Roberto Navigli. 2021. Named entity recognition for entity linking: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2584–2596.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Kehang Wang, Qi Liu, Kai Zhang, Ye Liu, Hanqing Tao, Zhenya Huang, and Enhong Chen. 2023. Class-dynamic and hierarchy-constrained network for entity linking. In *International Conference on Database Systems for Advanced Applications*, pages 622–638. Springer.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

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)*, pages 6397–6407.

Zilin Xiao, Ming Gong, Jie Wu, Xingyao Zhang, Linjun Shou, and Daxin Jiang. 2023. Instructed language models with retrievers are powerful entity linkers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2267–2282, Singapore. Association for Computational Linguistics.

Ziyun Xu, Chengyu Wang, Minghui Qiu, Fuli Luo, Runxin Xu, Songfang Huang, and Jun Huang. 2023. Making pre-trained language models end-to-end few-shot learners with contrastive prompt tuning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 438–446.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2021. Entqa: Entity linking as question answering. In *International Conference on Learning Representations*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

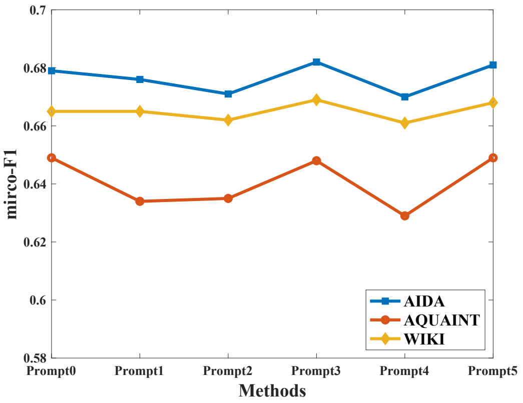


Figure 7: The Impact of Different Instruction Prompts

Dateset	ACE2004	AIDA	AQUAINT	CWEB	MSNBC	WIKI
GLM-8K	1.78	0.97	1.27	0.92	1.19	0.97
GLM-32K	1.88	1.13	1.58	1.12	1.31	1.24
Zephyr	15.22	11.19	15.69	13.08	14.75	14.26

Table 4: Execution Time (s) per Mention with Various Base Models.

A Experimental Supplement

A.1 Different Instruction Prompts

Due to space constraints, Figure 7 mentioned in the main text have been moved to the appendix, which shows the performance of the contextual linker with different instruction prompts in Section 5.5.3

A.2 Repeated Answers

To provide a more intuitive illustration of the robustness of our framework, we provide a case study of repeated responses. As shown in Table 5, although the expressions of the model outputs are different, none of the semantics of the results change, which demonstrates the stability of our framework.

A.3 Case Study for Prior

As we mention in Section 5.6, we also provide another case for prior linking in Figure 8. The entity mentioned is *Orange County*, which appears in an airport blog. Initially, the Entity Reduction Processor filtered out 23 irrelevant entities, such as *Orange County, Florida*, *Orange County (film)*. After this process, three relevant entities remained: *Orange County*, *California*, *John Wayne Airport*. Under the classification of *geography and places*, the contextual linker pinpointed *John Wayne Airport* as the probable reference, which thinks that the article's focus is on airports and travel. In contrast, the prior linker posited *Orange County, Cal-*

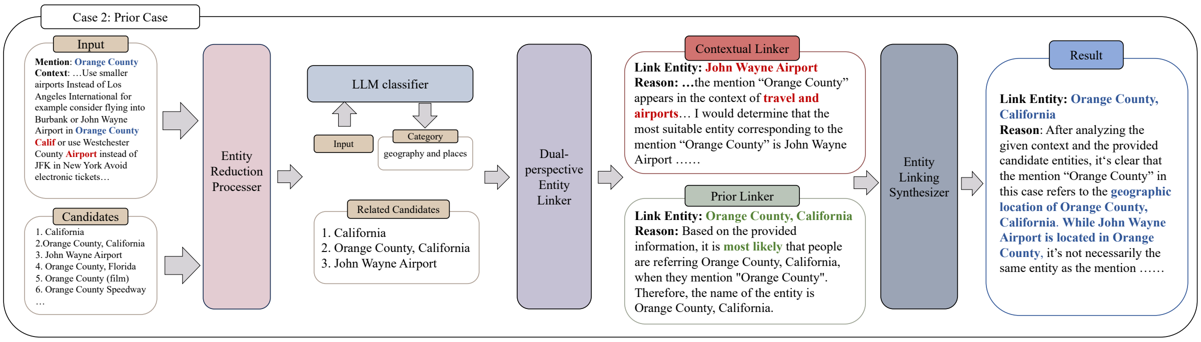

Case 2: Prior Case
Input
Mention: Orange County
Context: …Use smaller airports Instead of Los Angeles International for example consider flying into Burbank or John Wayne Airport in Orange County Calif or use Westchester County Airport instead of JFK in New York Avoid electronic tickets…
Candidates
1. California
2.Orange County, California
3. John Wayne Airport
4. Orange County, Florida
5. Orange County (film)
6. Orange County Speedway
…
Entity Reduction Processer
LLM classifier
Input
Category
geography and places
Related Candidates
1. California
2. Orange County, California
3. John Wayne Airport
Dual-perspective Entity Linker
Contextual Linker
Link Entity: John Wayne Airport
Reason: …the mention "Orange County" appears in the context of travel and airports… I would determine that the most suitable entity corresponding to the mention "Orange County" is John Wayne Airport ……
Prior Linker
Link Entity: Orange County, California
Reason: Based on the provided information, it is most likely that people are referring Orange County, California, when they mention "Orange County". Therefore, the name of the entity is Orange County, California.
Entity Linking Synthesizer
Result
Link Entity: Orange County, California
Reason: After analyzing the given context and the provided candidate entities, it's clear that the mention "Orange County" in this case refers to the geographic location of Orange County, California. While John Wayne Airport is located in Orange County, it's not necessarily the same entity as the mention ……

Figure 8: Case Study of OneNet. Prior with LLMs makes the prior linker reason correctly.

Mention	**Tallahassee**
Context	**Tallahassee** United States 11 15 AFP The Supreme Court in Florida today Wednesday refused the application by the state s authorities to ...
Entity	**Tallahassee, Florida**
Answer 1	Based on the given context and the features to consider for entity disambiguation, the best match for the mention Tallahassee appears to be entity **Tallahassee, Florida**. Here are the reasons for the selection:...
Answer 2	Based on the context provided and considering the features mentioned, the most suitable entity for the mention Tallahassee is entity **Tallahassee, Florida**. The context describes Tallahassee as the location where the Supreme Court in Florida refused to stop the new hand count of votes...
Answer 3	In this case, both candidate entities (Tallahassee, Florida and Tallahassee metropolitan area) are locations, but the mention Tallahassee in the given context is more likely referring to the capital city of Florida rather than the metropolitan area ... Therefore, the serial number for the correct entity is **"Tallahassee, Florida"**
Answer 4	Based on the context provided, it is most likely that the mention Tallahassee refers to entity **Tallahassee, Florida**. The context mentions the Supreme Court in Florida refusing an application to stop a vote count in some counties ...
Answer 5	Based on the context provided and considering the features mentioned, the most suitable entity for the mention Tallahassee is entity: **Tallahassee, Florida**. The context states that the Supreme Court in Florida ...

Table 5: Case Study of Repeated Answers

ifornia as the more frequent referent in general discourse. Ultimately, the Entity Consensus Judger gives precedence to the prior linker's prediction of *Orange County, California*. It noted that *John Wayne Airport* is situated within *Orange County, California*, which clarified the confusion for the contextual linker.

A.4 Framework Efficiency

As mentioned in Limitation, we acknowledge that the execution efficiency of the framework is indeed influenced by the inference speed of the base model. In order to address this, we have conducted performance evaluations and execution time measurements of our framework on various base models including Zephyr and GLM. The EL results are shown in Table 2, while the running time analysis is illustrated in Table 4. The execution time is obtained by randomly sampling 100 tests on each dataset without any parallelization acceleration.

In general, although the GLM model exhibits slightly lower performance compared to Zephyr, its inference speed surpasses it by up to tenfold. Furthermore, as we have mentioned in Limitation, there is ongoing research dedicated to optimizing inference speed for LLMs, which we believe can further enhance the efficiency of our framework.

B Framework Generalization

As discussed in Section 1, traditional methods heavily depend on external data such as entity priors,

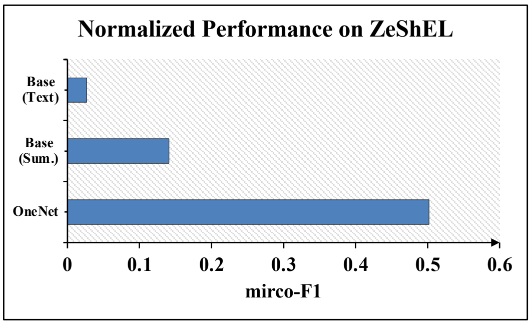


Figure 9: Results of Few-shot Entity Linking of Zeshel. Llama3 as the base model for larger context windows.

preventing their adaptability across different scenarios (Le and Titov, 2018). Moreover, these methods struggle with practical issues such as entity ID mapping, further complicating their migration across various knowledge bases (Tedeschi et al., 2021). In contrast, our framework, illustrated in Figure 2, leverages large language model textual reasoning and requires no fine-tuning, which enables our model to perform entity linking across diverse domains and knowledge bases.

To further discuss the generalizability of our framework, we present the normalized performance on the ZeShEL (Logeswaran et al., 2019) dataset in Figure 9. ZeShEL is an entity linking dataset constructed using Wikias from Fandom. We compare the performance of OneNet with its corresponding base model. For the base model, we used the settings described in Section 5.3. However, due to the excessive length of text in ZeshEL, the base model with original entity text resulted in poor performance. To address this, we also report the base model performance with LLM-generated entity summary, which reduces the context length.

In Figure 9, our framework achieves an improvement of approximately 35% over the base model, which shows the effectiveness of our approach. However, a performance gap remains compared to the results reported in Table 2 for the wiki-based dataset. We attribute this disparity partially to the influence of non-wiki data, but more significantly to the excessive length of ZeShEL's text. As shown in Table 6, ZeShEL surpasses other datasets in terms of the number of candidates, entity descriptions, and contexts, especially the contexts. The irrelevant information in excessively long contexts can mislead LLMs (Shi et al., 2023). To address this issue, we propose extracting critical information from contexts, such as the first and last sentences of paragraphs and sentences containing mentions, to enhance the performance of our framework.

Dateset	**ACE2004**	**AIDA**	**AQUAINT**	**CWEB**	**MSNBC**	**WIKI**	**ZeShEL**
Cand. Nums	42.25	7.18	34.69	6.98	25.93	6.09	**55.20**
Ent. Tokens	190.79	262.16	197.00	239.36	198.81	227.04	**441.65**
Cont. Tokens	171.17	452.53	169.57	222.126	198.88	195.90	**2394.93**

Table 6: Average of Candidates Numbers, Entity Description Tokens, and Context Tokens across Datasets.

C Prompt

In order to understand more intuitively how we prompted the different modules, Tables 7 and 8 show example prompts for all the modules and the specific context of the entity link instructions prompts respectively. Specifically, most of the prompts in Table 7 were written by hand to achieve the functionality we wanted, while most of the prompts in Table 8 were generated by GPT to distill knowledge from stronger models.

Module	Prompt
Summarization	The following is a description of {mention}. Please extract the key information of {mention} and summarize it in one sentence: {description}
Point-wise EL	You're an entity disambiguator. I'll give you the description of entity disambiguation and some tips on entity disambiguation, and you need to pay attention to these textual features: {Instruction Prompt}. Now, I'll give you a mention, a context, and a candidate entity, and the mention will be highlighted with '###'. Mention:{mention}, Context:{context}, Candidate Entity:one candidate entity. You need to determine if the mention and the candidate entity are related. Please refer to the above tips and give your reasons, and finally answer 'yes' or 'no'. Answer 'yes' when you think the information is insufficient or uncertain.
Category	You are a mention classifier. Wikipedia categorizes entity into the following categories: Categories. Now, I will give you a mention and its context, the mention will be highlighted with '###'. Mention:{mention}, Context:{context}. please determine which of the above categories the mention mention belongs to?
Contextual EL	You're an entity disambiguator. I'll give you the description of entity disambiguation and some tips on entity disambiguation, you should pay attention to these textual features: {Instruction Prompt}. The following example will help you understand the task: {CoT Prompt}. Now, I'll give you a mention, a context, and a candidate entity, and the mention will be highlighted with '###'. Mention:{mention}, Context:{context}, {Candidates} .You need to determine which candidate entity is more likely to be the mention. Please refer to the above example, give your reasons, and finally answer serial number of the entity and the name of the entity. If all candidate entities are not appropriate, you can answer '-1.None'.
Prior EL	You're an entity disambiguator. I'll provide you a mention and its candidates below. mention:{mention}. {Candidates}. Based on your knowledge, please determine which is the most likely entity when people refer to mention "{mention}", and finally answer the name of the entity.
Merge	You're an entity disambiguator. I'll give you the description of entity disambiguation and some tips on entity disambiguation, you should pay attention to these textual features: {Instruction Prompt}. Now, I'll give you a mention, a context, and a candidate entity, and the mention will be highlighted with '###'. Mention:{mention}, Context:{context}, {Candidates} .You need to determine which candidate entity is more likely to be the mention. Please refer to the above example, give your reasons, and finally answer serial number of the entity and the name of the entity. If all candidate entities are not appropriate, you can answer '-1.None'.

Table 7: Examples of Prompt for Each Module

Id	Context
Prompt 0	Entity Disambiguation Task: You will be given a context, a mention, and a set of candidate entities. Your goal is to identify the entity that corresponds to the mention within the context. Follow these steps:1. Read the context and identify the mention.2. Examine the candidate entities provided.3. Consider the following features to determine the best match for the mention:a. Categories: Look for category labels or descriptions that align with the mention.b. Modifiers: Pay attention to qualifying words that provide additional information about the mention.c. Contextual clues: Analyze the surrounding text for related entities, events, or relationships.d. Semantic meaning: Consider the meaning, context, and purpose of the mention and candidate entities. 4. Make an informed decision based on the available information and select the most suitable entity.
Prompt 1	Here's a hint to help your friend understand entity disambiguation and some features to consider:Entity disambiguation involves determining if a given candidate entity is the same as the mention within a given context. To make an accurate judgment, consider the following features:1. Categories: Look for clues indicating the category or type of the mention and the candidate entity. Are they both people, places, organizations, or something else? Matching categories often indicate a higher likelihood of being the same entity.2. Modifiers: Pay attention to descriptive words or phrases that modify the mention and the candidate entity. Do they share similar modifiers? For example, if the mention is 'red apple' and the candidate entity is 'juicy apple,' the shared modifier 'apple' suggests a potential match. 3. Contextual information: Analyze the surrounding text to understand the context in which the mention and candidate entity appear. Look for additional information that can help determine if they refer to the same entity. Consider factors such as location, time, relationships, or events mentioned. 4. Unique identifiers: Check for any unique identifiers associated with the mention and the candidate entity. These could be specific names, titles, dates, or other distinct attributes. Matching unique identifiers can strongly indicate a match. 5. Disambiguation cues: Look for disambiguation cues within the context that explicitly clarify or distinguish between different entities. These cues may include pronouns, definite or indefinite articles, or explicit references to other entities. Remember, entity disambiguation can sometimes be challenging, especially when dealing with ambiguous or incomplete information. It's important to carefully analyze the given context and consider multiple features to make an informed decision.
Prompt 2	Entity disambiguation is a common task in natural language processing (NLP) and information retrieval. The goal is to determine which specific entity is being referred to in a text when there may be multiple entities with the same or similar names. Here are some hints and features to look out for when you're doing this task manually: Context: The surrounding sentence or paragraph where the mention is located can provide clues about the entity. For example, if the mention is 'Apple' and the context is about technology or smartphones, it's likely referring to the technology company. If the context is about fruit or food, it's probably referring to the fruit. Categories: Entities often belong to specific categories or types, such as people, organizations, locations, etc. If you know the category of the candidate entity, this can help you decide if it matches the mention. For example, if the mention is 'Washington' and the candidate entity is a person (e.g., George Washington), but the context is about places, then the candidate entity is probably not a match. Modifiers: These are words or phrases that modify or add details to the mention. For example, in the mention 'President Obama,' the modifier 'President' indicates that the entity is a person, specifically Barack Obama. Modifiers can also include adjectives, descriptive phrases, or other context that helps specify the entity. Co-references: These are other mentions of the same entity in the text. If the text refers to 'Apple' multiple times and talks about both smartphones and fruit, you might be able to determine which 'Apple' is being referred to based on how it's discussed elsewhere in the text. Temporal and Geographical Factors: The time and place that the text was written can also provide clues. For example, if the mention is 'Jordan' in an article written in the 1990s about basketball, it's likely referring to Michael Jordan. If it's in a recent article about

Id	Context
	Middle Eastern politics, it's probably referring to the country Jordan. External Knowledge: Sometimes, you might need to use knowledge that's not contained in the text. For example, if the mention is 'Musk' and the context is about space travel, you might need to know that Elon Musk is the CEO of SpaceX to realize that 'Musk' refers to him. Remember, entity disambiguation can be tricky, and there might not always be a clear answer. It often requires a combination of understanding the text, knowing about the world, and using your best judgment.
Prompt 3	Context: Look at the surrounding text to understand the topic.Categories: Consider the type of the entity (person, organization, location, etc.).Modifiers: Pay attention to words or phrases that add details to the mention.Co-references: Check other mentions of the same entity in the text.Temporal and Geographical Factors: Consider when and where the text was written.External Knowledge: Use outside knowledge not contained in the text.Remember, entity disambiguation requires understanding the text, knowing about the world, and using good judgment.
Prompt 4	**Entity Disambiguation Task** Objective: Your goal is to identify the correct entity from a list of candidate entities that corresponds to a given mention within a specific context. Procedure: You will be provided with three things: 1. Context: This is a paragraph or a set of sentences that provides the surrounding information where the mention is found. 2. Mention: This is the specific term or phrase that you need to disambiguate – i.e., to identify its correct meaning or reference. 3. Candidate Entities: This is a list of possible entities that the mention could refer to. Your job is to select the correct one based on the context. Features to Look Out For: 1. **Categories/Types**: Entities belong to different categories such as people, organizations, locations, events, etc. The category of the mention can often be inferred from the context. For instance, if the context is discussing a concert, the mention is likely referring to a musician or a music-related entity. 2. **Modifiers**: These are words or phrases that provide additional information about the mention. For example, in the mention 'Apple CEO Tim Cook', 'Apple CEO' is a modifier that helps distinguish this Tim Cook from other individuals with the same name. 3. **Co-references**: These are other mentions of the same entity in the context. They can provide additional clues about the entity. For example, if the context mentions 'the tech giant' before mentioning 'Apple', these two are co-references pointing to the same entity. 4. **Temporal and Spatial Clues**: The time and place mentioned in the context can also help in disambiguating the entity. For example, if the context is about the 19th century, a mention of 'Washington' is more likely to refer to George Washington than the city of Washington D.C. 5. **Domain-specific Knowledge**: Sometimes, general world knowledge or domain-specific knowledge can help disambiguate entities. For example, if the context is about computer programming, a mention of 'Python' is likely referring to the programming language, not the snake. Remember, the goal is to use the context and your understanding of the world to determine which entity from the list of candidates the mention is most likely referring to. It's not always easy, and there may be times when more than one candidate seems possible. In such cases, choose the one that seems most likely based on all the available information. Good luck!
Prompt 5	**Entity Disambiguation Task** Goal: Identify the correct entity from a list of candidates that matches a given mention within its context.Procedure: You'll get a context (surrounding text), a mention (term to identify), and candidate entities (possible matches). Key Features: 1. Categories: Check if the context implies a category (person, place, etc.) for the mention. 2. Modifiers: Look for additional info (e.g., 'Apple CEO Tim Cook') that distinguishes the mention.3. Co-references: Find other mentions of the same entity in the context for extra clues.4. Temporal/Spatial Clues: Time and place details can help disambiguate the entity.5. Domain Knowledge: Use general or specific knowledge to infer the correct entity. Use all available information to select the most likely entity from the candidates. Good luck!

Table 8: The Instruction Prompts of Entity Linking Generated by GPT