Entity Disambiguation via Fusion Entity Decoding

Anonymous ACL submission

Abstract

Entity disambiguation (ED), which links the mentions of ambiguous entities to their referent entities in a knowledge base, serves as a core component in entity linking (EL). Existing generative approaches show better accuracy compared to classification approaches on conventional benchmarks. Nevertheless, generative approaches suffer from large-scale pre-training and inefficient generation. Most importantly, entity descriptions, which could contain crucial information to distinguish similar entities from each other, are often overlooked. We propose an encoder-decoder model to disambiguate entities with more detailed entity descriptions. Given text and candidate entities, the encoder learns interactions between the text and each candidate entity, producing representations for each entity candidate. The decoder then fuses the representations of entity candidates together and select the correct entity. Our experiments, conducted on various entity disambiguation benchmarks, demonstrate the strong and robust performance of this model, particularly +1.5% in the ZELDA benchmark compared with GENRE. Furthermore, we integrate this approach into the retrieval/reader framework and observe +1.5% improvements in end-to-end entity linking in the GERBIL benchmark compared with EntQA.

1 Introduction

Entity linking (EL) extracts references (a.k.a. mentions) to entities within a document and associates these mentions with their corresponding entries in a knowledge base (KB). EL is a fundamental component in automatic text comprehension, with various practical applications such as question answering, text analysis, recommender systems, semantic search, and information retrieval.

As the most critical component of EL workflows, entity disambiguation (ED) aims to select the correct entity from a set of candidate entities, given textual references. For instance, the entity mention 'Bert' may stand for 'the famous language model' (Devlin et al., 2018) or 'the golden yellow Muppet character' depending on the given context. Therefore, models need to understand context to disambiguate entities correctly.

Owing to its practical significance in the industry and the latest developments in utilizing pre-trained language models (Devlin et al., 2018; Lewis et al., 2020; Liu et al., 2019; Raffel et al., 2020), various approaches for entity disambiguation have been introduced in recent years. Primarily, existing methods can be categorized into two styles: classification approaches (Yamada et al., 2022; Févry et al., 2020) or generative approaches (De Cao et al., 2021). Classification approaches such as (Yamada et al., 2022) predict the masked entity titles while generative approaches such as (De Cao et al., 2021) directly decode entity titles.

The recently proposed ZELDA benchmark (Milich and Akbik, 2023) standardizes the experimental setup (consistent training data, entity vocabulary, and candidate lists) and shows that generative approaches such as (De Cao et al., 2021) have significantly stronger performance under this experimental setup.

However, authors in (Zhang et al., 2022) argue that generative approaches require large scale pretraining. In particular, (De Cao et al., 2021) critically relies on a prefix tree (also known as a trie) derived from Wikipedia to constrain the beam search in order to produce a valid entity title in a given knowledge base (KB), which might be inefficient memory-wise. In addition, since it directly generates a valid entity without reading their descriptions, crucial information in the descriptions might be ignored. Therefore, disentangling significantly similar entities proves challenging with this method (Milich and Akbik, 2023).

To better disentangle similar entities, in this paper we propose an encoder-decoder model that decodes entities by utilizing their descriptions. Our approach is mainly inspired by a recent work on question answering (Izacard and Grave, 2021). In particular, we make the following contributions: We summarize our contributions in the following:

- We propose a new ED approach, using an encoder-decoder model. Given text and entity candidates, the encoder learns the interactions between the text and each entity candidate, generating representations for each candidate. Subsequently, the decoder fuses these candidate entity representations and generates correct entities. At inference, instead of relying on a constrained beam search, it only needs simple greedy decoding.

- We follow the standard evaluation practice (ensuring consistent knowledge base, training corpus and entity candidate lists) and rigorously evaluate this approach in several ED benchmarks (Milich and Akbik, 2023) and show its strong and robust performance.

- We integrate our approach into an end-to-end entity linking pipeline and show large improvements compared with the current state-of-the-art in GERBIL (Usbeck et al., 2015) benchmark. To the best of our knowledge, our approach is the first retrieval-augmented generation approach in EL.

- We propose retrieval augmented entity linking using Large Language Models (LLMs), e.g., GPT-4 and evaluate it in GERBIL (Usbeck et al., 2015) benchmark. Our results show that with augmented entity retrieval, GPT-4 outperforms the current SoTA on some datasets but in general, it underperforms compared to fine-tuning-based approaches.

Our approach outperforms strongest ED baselines (De Cao et al., 2021; Févry et al., 2020; Yamada et al., 2022) on ZELDA benchmark and EL baselines (De Cao et al., 2021; Zhang et al., 2022; Shavarani and Sarkar, 2023) on GEBIL benchmark (Usbeck et al., 2015).

2 Related Work

Entity Disambiguation. Existing ED approaches typically fall into two main categories: classification approaches and generative approaches.

For classification approaches, LUKE (Yamada et al., 2022) and FEVRY (Févry et al., 2020) are two of the most well-known approaches due to their strong performance. LUKE is based on masked entity prediction. To be more specific, during the pre-training, LUKE combines input text and ground-truth entities as input tokens. Then, it randomly masks entities from those ground-truth entities and predict those masked entities by leveraging both the input text and those unmasked entities. Their model is trained on a large entity-annotated corpus obtained from Wikipedia and achieves the current SoTA in several ED benchmark datasets.

For generative approaches, GENRE (De Cao et al., 2021) uses BART weights from (Lewis et al., 2020) and is trained on a Wikipedia corpus, learning to generate entity names in an autoregressive manner, conditioned on the provided context. At inference, GENRE employs a constrained beam search strategy that forces each generated name to be in a predefined entity set.

Conventionally, ED methods are evaluated on six datasets, MSNBC, AQUAINT, ACE2004, WNED-CWEB (CWEB) and WNED-WIKI (WIKI) (Gabrilovich et al., 2013; Guo and Barbosa, 2018). Nevertheless, as shown in Milich and Akbik (2023), those different ED methods use significantly different amounts of training data (ranging from 2 to 20 million annotated text) obtained with diverse sampling methodologies and enhanced weak labels (Orr et al., 2020; Broscheit, 2020), and completely different knowledge bases (ranging from few thousands to over 6 million) from different sources, YAGO (Suchanek et al., 2007) or KILT (Petroni et al., 2021) and different candidate lists (Hoffart et al., 2011; Pershina et al., 2015). Thus, comparing various approaches is highly challenging. It is impossible to conclude which approach performs best (Milich and Akbik, 2023).

ZELDA (Milich and Akbik, 2023) benchmark is proposed to unify the training data set, entity vocabulary, and candidate lists to facilitate direct comparability of ED approaches. For this reason, we compare our approach with SoTA approaches on ZELDA benchmark. Our experiment is rigorously conducted using the same training data, entity vocabulary, and candidate lists without additional information from Wikipedia or using weak labels.

Entity Linking. Different from ED, the key challenge of EL is its significantly large search space. A system can potentially generate any subset of conceivable spans in the document, each of which could correspond to an entity in a large KB, typically containing millions of entities. To manage this overwhelming scale, existing approaches break down EL into two stage tasks: mention detection (MD) and entity disambiguation (ED). These tasks are often tackled with varying degrees of independence.

In most of these approaches, the sequence of subproblems is consistent: first, the system identifies possible entity mentions, and then it links these mentions to specific entries in the given knowledge base. This MD→ED classic pipeline is utilized in most methods. They either assume that mentions are provided in advance, following the example of Gupta et al. (2017) or take a different route by employing readily available entity recognition systems to first identify mentions and then disambiguate them through the ED process, as evidenced in the works of Hoffart et al. (2011); Li et al. (2020). Furthermore, some research (Kolitsas et al., 2018; De Cao et al., 2021) trains an end-to-end autoregressive model that jointly performs MD→ED by beam search.

Recently, (Zhang et al., 2022) has shown that the classic MD (i.e., retrieval) → ED (i.e., reader) approach suffers from identifying mentions without prior knowledge of their corresponding entities, which is unnatural and challenging. To fix this problem, the authors flip the order of MD and ED, and propose an ED → MD pipeline. Their key observation is that finding relevant candidate entities is easy without the knowledge of their specific mentions. Their ED → MD approach achieves SoTA results on the in-domain AIDA-CoNLL dataset (Hoffart et al., 2011) and GERBIL benchmark (Usbeck et al., 2015). Although their retriever (select top-k candidate entities) performs remarkably well, the majority of errors are attributed to their reader (which predicts the final entities and mention spans).

A recent work (Shavarani and Sarkar, 2023) proposes a structured prediction approach and achieves 88.6% on AIDA-CoNLL test-b by using the PPRforNED (Pershina et al., 2015) candidate list. However, Yang et al. (2018); Milich and Akbik (2023) question this candidate list since it is unclear how candidates were pruned. The entity candidates generated by PPRforNED (Pershina et al., 2015) were found to be well-tailored to the AIDA-CoNLL test-b evaluation dataset, with high recall and low ambiguity. Models (Yamada et al., 2022; Févry et al., 2020) improve significantly when using these lists instead of the more generic lists by (Hoffart et al., 2011) and (Ganea and Hofmann, 2017), respectively. Without the handcrafted PPRforNED (Pershina et al., 2015) candidate list, the result of ADIA-CONLL test-b in (Shavarani and Sarkar, 2023) is the same as (Zhang et al., 2022), 85.8%.

As discussed in ZELDA (Milich and Akbik, 2023), using additional signals makes comparison unfair and indirect. Moreover, in real world entity linking applications, additional signals such as pruned candidate lists may not be available. Therefore, same as our comparison methodology in ED, we do not bring any additional signals and aim to conduct an end-to-end direct entity linking comparison precisely by using the same training data and same knowledge base, KILT (Petroni et al., 2021) as EntQA (Zhang et al., 2022) and GENRE (De Cao et al., 2021).

3 Model

3.1 Entity Disambiguation

We formalize the ED task as follows. Given a set of candidate entities denoted as $\mathcal{E}$ in a Knowledge Base (KB), and an input text D with a single mention flagged with two special start token and end token, the goal is to find the proper entity $e \in \mathcal{E}$ that corresponds to the mention in D.

In Figure 1, we show an example of entity disambiguation. Given a text with annotated mention that represents what we want to disambiguate, we add special tokens <s1> and <e1> before and after the mention to denote the corresponding mention that we want to disambiguate. We concatenate input text with information from each entity candidate including entity title and entity description, and feed it into the encoder model to form an entity representation and the decoder model takes the fused entity representations from all those candidates to generate the correct entity name.

3.2 Entity Linking

We formalize the EL task as follows. Given a set of entities denoted as $\mathcal{E}$ in a Knowledge Base (KB), and an input document D, the objective is to identify every entity $e \in \mathcal{E}$ along with a mention m such that $m \in D$ and m links to e. Typically, the length of D varies from few words (e.g., short queries) to few thousands of words (e.g., news). To handle long document entity linking, previous

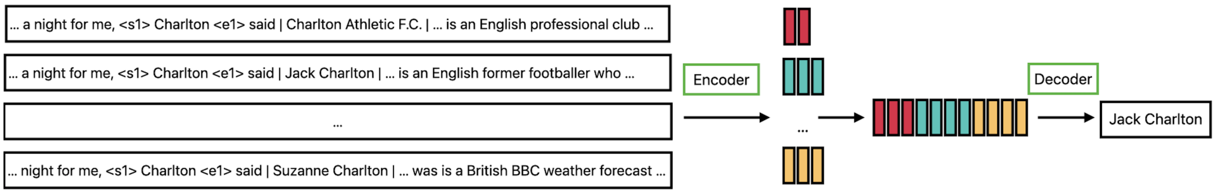

Figure 1: Pipeline of the fusion entity decoding for entity disambiguation. Given a text *'DUBLIN 1996-12-07 Jack Charlton's relationship with the people of Ireland was cemented on Saturday when the Englishman was officially declared one of their own. (few sentences are abbreviated here) That is why this is so emotional a night for me , <s1> Charlton <e1> said'*. Follow (De Cao et al., 2021), we add special tokens <s1> and <e1> to denote the corresponding mention to disambiguate. Given candidate entities 'Charlton Athletic F.C.', 'Jack Charlton', 'Bobby Charlton', 'Suzanne Charlton' from KB, we concatenate text with each entity candidate, including its entity title and its description. The Encoder learns interactions between the text and each entity candidate and produces suitable representations for each entity candidate; decoder concatenates those representations and selects the correct entity.

research (Zhang et al., 2022) typically segments each document D into sentence chunks. For each sentence chunk p, most approaches (Hoffart et al., 2011; Li et al., 2020) commonly break down the task of EL for a sentence chunk p into two main components: mention detection (MD) and entity disambiguation (ED), and first extract mentions from passages (MD) and then link to entities (ED).

Zhang et al. (2022) introduce a different two-stage process, instead of first identifying mentions and then link them entities, it first retrieve top-k candidate entities, followed by the reader's task of picking up the accurate entities along with predicting their associated mention spans. Figure 2 illustrates an instance of end-to-end EL employing the retrieval-plus-reader approach. Our approach follow this pipeline.

3.2.1 Bi-encoder EL Retrieval

Entity Embedding. Following (Wu et al., 2019), we represent an entity e as a combination of its title and description using the format: `[CLS] title(`e`) [ENT] description(`e`) [SEP]`. `[ENT]` is a special token to separate the entity title and description representation. For Wikipedia entities, we consider up to 128 tokens for their descriptions. We use an encoder $\mathbf{enc}_E$ to produce an embedding for an entity e.

Passage Embedding. For each passages p with its document topics t, we also concatenate those information using the following format: `[CLS]` p `[SEP]` t `[SEP]`. We use another encoder $\mathbf{enc}_P$ to produce an embedding for a passage p.

Training. The score of an entity e and a passage p is given as $s(e, p) = \mathbf{enc}_E(e)^\top \mathbf{enc}_P(p)$. Same as (Zhang et al., 2022), we train the retriever using a multi-label variant of noise contrastive estimation (NCE) (Zhang and Stratos, 2021).

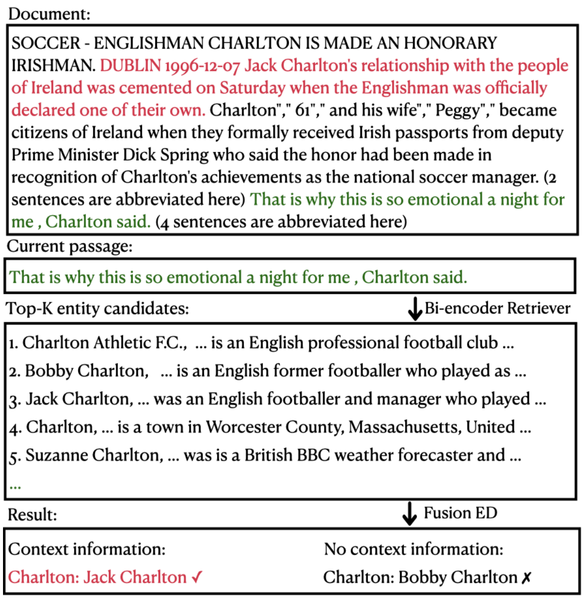

Figure 2: Example of document level entity linking from AIDA test. Given a document, FusionED splits it into smaller passage chunks. Given the current passage 'That is why this is so emotional a night for me, Charlton said.', the bi-encoder entity retrieval picks up top 100 entity candidates, e.g., 'Charlton Athletic F.C.', 'Bobby Charlton', 'Jack Charlton'. FusionED then decodes linked entities and mentions using entity candidate lists.

3.2.2 Fusion EL Reader

We use a similar architecture to the one we used for ED (Figure 1), while the model generates both entity names and mentions instead of only generating entity names as this was the case in ED.

Given a passage chunk p along with its truncated original document D, the retrieval retrieves the top-k candidate entities $e_1, \cdots, e_k$.

Then, for each retrieved candidate entity e_i, we concatenate the document D, the current passage chunk p, the entity title of e_i, and the entity description of e_i. We add special tokens <extra_id_0>, <extra_id_1>, <extra_id_2>, <extra_id_3> before the document, the current passage chunk, the entity name, and the entity description, respectively. The input format becomes <extra_id_0> D <extra_id_1> p <extra_id_2> title(e_i) <extra_id_3> description(e_i).

The encoder independently processes input data for each entity candidate e_i and then merges the resulting representations from all the candidates. Finally, the decoder performs the attention over the merged representations of all the retrieved entities. If no candidate entities are linked, the decoder output an empty string. Otherwise, for each linked entity e_i, it outputs e_i <extra_id_4> $m_{i1}, \cdots, m_{in}$ where $m_{i1}, \cdots, m_{in}$ are all mentions from p which links to e_i. Finally, we use a special token <extra_id_5> to split the decoding output from each entity e_i. Therefore, the final output sting is e_1 <extra_id_4> $m_{11}, \cdots m_{1n}$ <extra_id_5> e_2 <extra_id_4> $m_{21}, \cdots m_{2n}$ <extra_id_5> $\cdots$ e_i <extra_id_4> $m_{i1}, \cdots, m_{in}$.

4 Experiment

We conduct extensive experiments to demonstrate the performance of our proposed approach (FUSIONED) over 20 datasets, addressing both single-entity disambiguation and end-to-end entity linking. The goal of our experiments is to facilitate a direct comparison, illustrating that under identical conditions (without incorporating extra training data or taking additional signals into account), our approach outperforms the current SoTA.

4.1 Entity Disambiguation

Setup. We follow the experiment setup in ZELDA benchmark (Milich and Akbik, 2023), using their training data, entity vocabulary and the more generic candidate list. We initialize the weights of our model using FLAN-T5-base (Chung et al., 2022) 220M to match the number of parameters of SoTA models (274M for LUKE (Yamada et al., 2022) and FEVRY (Févry et al., 2020), 178M for GENRE (De Cao et al., 2021)). We train the model for 60k steps with a learning rate 0.0001 using Adam optimizer (Kingma and Ba, 2015), with a batch size of 12 on 12 NVIDIA Tesla V100 32GB.

Given a context with a mention, we consider approximately 250 tokens surrounding the annotated mention. For each entity candidate, we concatenate the entity name, a special token, and the entity description, truncating to a maximum of 140 tokens. Then, for each context, we utilize the candidate list from the benchmark (Milich and Akbik, 2023). We only consider the top 200 entity candidates from this list. We evaluate checkpoints every 2000 steps for the last 8000 steps in AIDA-B, selecting the best checkpoint.

Datasets. At inference, we evaluate the model using greedy decoding on 9 datasets: AIDA-B (Hoffart et al., 2011), TWEEKI (Botzer et al., 2021), REDDIT-POSTS and REDDIT-COMMENTS (Botzer et al., 2021), WNED-WIKI and WNED-CWEB (Guo and Barbosa, 2018), SLINKS-TOP and SLINKS-SHADOW and SLINKS-TAIL (Provatorova et al., 2021). These datasets are collected from diverse sources: news (AIDA-B), annotated tweets (TWEEKI), top-scoring Reddit posts and comments (REDDIT-POSTS and REDDIT-COMMENTS), Wikipedia articles (WNED-WIKI and WNED-CWEB). In particular, (Provatorova et al., 2021) categorizes entities into three cases based on their appearance frequency in Wikipedia: SLINKS-TOP, where the ground truth entity is the most frequent; SLINKS-SHADOW, where a more popular entity overshadows the correct disambiguation; and SLINKS-TAIL, for rare long-tail entities.

Baselines. We examine two methods presented in (Févry et al., 2020) using a candidate list (FEVRY_{CL}) and without any restriction on the search space ($\text{FEVRY}_{\text{ALL}}$). Additionally, for one of the ED SoTA approaches LUKE (Yamada et al., 2022), we present results of two models $\text{LUKEP}_{\text{PRE}}$ and LUKEP_{FT} on ZELDA (Milich and Akbik, 2023) benchmark.

GENRE (De Cao et al., 2021) employs a prefix tree derived from all entity titles in the KB to restrict the generation process. While GENRE does not utilize candidate lists during training, in inference the prefix tree can be generated using the candidate lists GENRE_{CL} or without candidate lists $\text{GENRE}_{\text{ALL}}$.

We also list CL-RECALL, which is the recall

Method	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COMM	WNED-CWEB	WNED-WIKI	SLINKS-TAIL	SLINKS-SHADOW	SLINKS-TOP	AVG
Baselines										
CL-RECALL	91.1	94.0	98.4	98.3	92.4	98.8	98.8	56.7	73.1	89.1
Classification										
FEVRY$_{\text{ALL}}$	79.2	71.8	88.5	84.1	68.0	84.3	63.8	**43.4**	53.1	70.7
FEVRY$_{\text{CL}}$	79.5	76.9	89.0	86.5	70.3	84.5	87.6	31.9	47.7	72.7
LUKEP$_{\text{PRE}}$	79.3	73.8	76.1	69.9	66.8	68.4	97.7	20.4	50.8	67.0
LUKEP$_{\text{FT}}$	**81.2**	77.9	81.5	78.5	70.3	76.5	98.0	22.5	51.8	71.0
Generative										
GENRE$_{\text{ALL}}$	72.4	75.9	88.8	83.9	66.5	85.2	95.3	38.7	43.5	72.2
GENRE$_{\text{CL}}$	78.6	80.1	92.8	91.5	**73.6**	88.4	**99.6**	37.3	52.8	77.2
FUSIONED	80.1	**81.4**	**93.9**	**92.3**	**73.6**	**89.0**	98.3	41.5	**57.9**	**78.7**

Table 1: Comparison between FusionED with both classification or generative based SoTA in ZELDA Benchmark (Milich and Akbik, 2023). Baselines number are taken from (Milich and Akbik, 2023). We emphasize the leading model by formatting it in **bold** and the second-best model by using an underline for each dataset. CL-RECALL represents the recall of the candidate list in ZELDA, indicating the highest possible accuracy using its candidate list.

of the candidate list in ZELDA. It reflects the best possible accuracy if we always select the correct entity from the candidate list.

Experimental Results. Table 1 reports the accuracy of FUSIONED compared with SoTA models. Clearly, FUSIONED achieves the highest performance across six datasets and secures the three position in two datasets. According to Table 1 and as it was previously pointed out by (Milich and Akbik, 2023), GENRE shows significantly better performance over classification-based baselines. However, it struggles to disambiguate entities in SLINKS-TOP and SLINKS-SHADOW. One possible interpretation is that it never uses any entity description to disambiguate entities with a similar title. Thus, it favors decoding into the most prominent case where the generated entity title will be most similar to the mention text.

It is worth mentioning that FUSIONED demonstrates an over +4 point accuracy improvement compared to GENRE on SLINKS-TOP and SLINKS-SHADOW datasets. These datasets involve ambiguous entities with similar titles. Incorporating information from entity descriptions is a prominent reason for FUSIONED's enhanced performance.

Table 2 shows the accuracy of different approaches across various difficulty brackets in the WNED-WIKI dataset. [0.4 - 0.3] represents the most difficult test cases while 1 represents the easiest ones. Our model has the highest accuracy across most different brackets (+5% in [0.4 - 0.3]), suggesting that using entity descriptions can help disambiguate closed entities in most challenging test cases. While it underperforms GENRE for easiest cases, possibly due to our model being trained with only 30% of the data, limited by computational constraints.

4.2 Entity Linking

Setup. For EL, we adhere to the established convention (De Cao et al., 2021; Zhang et al., 2022) by presenting the InKB Micro F1 score for both the in-domain and out-of-domain datasets. Specifically, for the in-domain scenario, we train FUSIONED using the AIDA-CoNLL dataset (Hoffart et al., 2011). For the out-of-domain tests, following the same practice, we evaluate it on seven test sets: MSNBC (Cucerzan, 2007), Derczynski (Der) (Derczynski et al., 2015), KORE 50 (K50) (Hoffart et al., 2012), N3-Reuters-128 (R128), N3-RSS-500 (R500) (Röder et al., 2014), and OKE challenge 2015 and 2016 (OKE15 and OKE16) (Nuzzolese et al., 2015). For KB, we utilize the 2019 Wikipedia dump, as supplied within the KILT benchmark (Petroni et al., 2021), encompassing a total of 5.9 million entities for our knowledge base (KB).

Retriever Training. Following (Zhang et al., 2022), we initialize weights of both the passage encoder ($\mathbf{enc}_P$) and the entity encoder ($\mathbf{enc}_E$) using BLINK (Wu et al., 2019) retrievers that have been pretrained on Wikipedia hyperlinks. We also fine-tune retrievers using NCE objective with hard negative mining and follow the same sampling strategy as (Zhang et al., 2022) (90% from random sample and 10% from hard negatives) . We reproduce their retriever by matching their top-100 recall numbers reported in their paper. We use FAISS (Johnson et al., 2019) to speed up vector similarity search.

Reader Training. We create the reader dataset by selecting the top 100 candidates from the retrieval process. For each ground truth entity, we

Method	1	[1 - 0.9]	[0.9 - 0.8]	[0.8 - 0.7]	[0.7 - 0.6]	[0.6 - 0.5]	[0.5 - 0.4]	[0.4 - 0.3]
CL-RECALL	99.7	97.2	99.2	98.3	98.3	99.1	98.8	99.6
FEVRY$_{CL}$	94.8	92.2	88.8	87.2	84.1	80.0	76.0	72.2
LUKEP$_{FT}$	91.5	90.4	86.3	80.3	77.8	73.8	62.2	56.2
GENRE$_{CL}$	**97.1**	**94.2**	**91.2**	85.6	87.8	86.9	86.9	79.7
FusionED	96.4	92.4	90.8	**87.5**	86.1	**88.1**	**87.1**	**85.0**

Table 2: Accuracy across various difficulty brackets was assessed for different approaches in the WNED-WIKI dataset. [0.4 - 0.3] is the most difficult bracket while 1 is the easiest. We emphasize the leading model by highlighting it in bold and denote the runner-up with an underline for each bracket. Our model shows the best performance across most different brackets, suggesting that using entity descriptions can help disambiguate closed entities in most challenging tests.

	In-domain	Out-of-domain							
Method	AIDA-B	MSNBC	Der	K50	R128	R500	OKE15	OKE16	AVG
Hoffart et al. (2011)	72.8	65.1	32.6	55.4	46.4	**42.4**	**63.1**	0	47.2
Steinmetz and Sack (2013)	42.3	30.9	26.5	46.8	18.1	20.5	46.2	46.4	34.7
Moro et al. (2014)	48.5	39.7	29.8	55.9	23.0	29.1	41.9	37.7	38.2
Kolitsas et al. (2018)	82.4	72.4	34.1	35.2	50.3	38.2	61.9	52.7	53.4
Broscheit (2019)	79.3	-	-	-	-	-	-	-	
Martins et al. (2019)	81.9	-	-	-	-	-	-	-	
Van Hulst et al. (2020)	80.5	72.4	41.1	50.7	49.9	35.0	**63.1**	**58.3**	56.4
De Cao et al. (2021)	83.7	**73.7**	54.1	60.7	46.7	40.3	56.1	50.0	58.2
De Cao et al. (2021)	85.5	-	-	-	-	-	-	-	
Zhang et al. (2022)	85.8	72.1	52.9	64.5	**54.1**	41.9	61.1	51.3	60.5
Shavarani and Sarkar (2023)	85.8	63.1	**59.1**	53.7	47.1	44.4	59.5	56.6	58.7
GPT-4 (zero-shot) Shavarani and Sarkar (2023)	54.1	-	-	-	-	-	-	-	
GPT-4 + retrieval (zero-shot)	58.4	42.4	40.1	**69.0**	35.1	29.4	58.3	53.1	48.3
GPT-4 + retrieval (zero-shot)*	59.1	42.5	41.0	67.6	36.4	30.1	58.4	53.0	48.5
FusionED	**86.5**	73.6	56.8	65.1	53.1	41.6	62.3	56.6	**62.0**

Table 3: InKB Micro F1 on the GERBIL benchmark with respect to in-domain and out-of-domain test sets. We highlight the top-performing model in **bold** and the runner-up in underline for each dataset. For (Shavarani and Sarkar, 2023), to make a fair comparison, we use their AIDA-testb result without external additional candidate set (Pershina et al., 2015). For GPT-4 + retrieval (zero-shot)*, we additionally filter entities generated by the model using candidate entities obtained from entity retrieval and this slightly improve its overall performance.

create an entity title and mention pair. The model is initialized with the FLAN-T5-large model (Chung et al., 2022). We finetune the model for 20k steps with a learning rate of 0.0001 using the Adam optimizer (Kingma and Ba, 2015), with a batch size of 8, employing 8 NVIDIA Tesla A100 40GB GPUs. Following the approach in (Zhang et al., 2022), we evaluate the models every 1000 steps in AIDA and select the best checkpoint. We use a linear decay learning rate scheduler that starts at 0, warms up to the peak learning rate, and then decays back to 0. The warm-up rate is set to 1%.

Inference. During inference, we employ a sliding window approach to split the document into passages with a window size of 20 tokens and a stride of 10 tokens to avoid cutting off any mentions. For each split passage, we first retrieve the top 100 entity candidates using the bi-encoder, followed by a FusionED reader to decode correct entities along with their mentions.

Experimental Results. Table 3 shows results of FusionED compared with different entity linking systems. Clearly, FusionED achieves the best in-domain test (+0.7% F1 for AIDA-B (Hoffart et al., 2011)) without using any handcrafted candidate list (Pershina et al., 2015)

Overall, FusionED achieves the best averaged F1 score across the all evaluation datasets; +1.5% over EntQA(Zhang et al., 2022) and +2.8% over the latest work (Shavarani and Sarkar, 2023) in EL. The reason for the lower performance on OKE15 and OKE16 (Nuzzolese et al., 2015) is consistent with the observation made by (De Cao et al., 2021): these datasets include coreference annotations (such as pronouns and common nouns linked to entities), for which our model lacks training. In contrast, many other systems incorporate a component in their pipelines specifically designed to use these annotations.

Compared to the previous retrieval-plus-reader approach, EntQA (Zhang et al., 2022), FusionED improves by +1.5% on MSNBC, +3.9% on Der, +0.6% on K50, and +4.7% on OKE16.

4.3 Case Study: Retrieval-augmented LLMs for Entity Linking

Datasets	GPT-4 + retrieval			FusionED		
	P	R	F1	P	R	F1
AIDA-B	52.0	66.6	58.4	**84.4**	**88.7**	**86.5**
MSNBC	32.6	60.7	42.4	**75.6**	**71.7**	**73.6**
Der	29.2	**63.9**	40.1	**55.2**	58.5	**56.8**
K50	70.3	**67.8**	**69.0**	**72.0**	59.4	65.1
R128	25.6	**55.6**	35.1	**56.3**	50.2	**53.1**
R500	19.2	**62.8**	29.4	**31.6**	60.7	**41.6**
OKE15	64.1	**53.5**	58.3	**80.1**	51.0	**62.3**
OKE16	60.7	**47.2**	53.1	**76.8**	44.8	**56.6**

Table 4: In contrast to FusionED, GPT-4 + retrieval demonstrates improved recall (R) across all datasets except AIDA-B and MSNBC, while exhibiting inferior precision (P) across all datasets.

(Shavarani and Sarkar, 2023) has benchmarked LLMs for EL using the approach introduced in (De Cao et al., 2021) where it produces a markup around the mentions followed by the linked entity name. However, the results are much worse than our approach, 54.1 vs 86.5. Although LLMs possess comprehensive knowledge about entities, they face a limitation in directly reasoning about specific Wikipedia URLs and Wikipedia names.

We conduct a preliminary study to assess the performance of retrieval-augmented prompting for linking entities using LLMs. This approach involves utilizing the same retrieval models that we described before, which are initialized using BLINK (Wu et al., 2019) weights and fine-tuned based on AIDA (Hoffart et al., 2011). For the reader, we replace the FusionED with GPT-4. More precisely, we provide GPT-4 with truncated documents (up to 50 tokens), input passages, and entity candidates, including entity title and entity description (up to 50 tokens). We prompt it to link entities from the candidate entity sets and identify their corresponding mentions. To the best of our knowledge, we are the first to propose retrieval-augmented LLMs for EL.

Table 4 presents a detailed comparison between FusionED and GPT-4 + retrieval. GPT-4 + retrieval shows better recall (R) in all datasets except MSNBC, but it has lower precision (P) in all datasets. The inferior precision of GPT-4 might stem from 1) ambiguity in defining entities, where it considers instances like ‘Spoon’, ‘Pasta’, ‘Scientist’ as entities diverge from actual ground truth labels in MSNBC (Cucerzan, 2007); 2) linking ambiguous partial names to famous entities (e.g., in a dataset based on tweets (Derczynski et al., 2015), a given query is ‘I’m going home to Wisconsin’, it links the ambiguous entity ‘Wisconsin’ to the Wisconsin state, but it may refer to ‘University of Wisconsin–Madison’). Our preliminary results suggest that future research should focus on enhancing the precision of LLMs by using varied prompts to match SoTA fine-tuned models.

5 Conclusion

We propose a an encoder-decoder model architecture to enhance the disambiguation of entities by providing more detailed descriptions. The encoder, when given text and candidate entities learns the interactions between the text and each entity candidate, generating representations for each candidate. The decoder then combines these representations to produce the correct entity. Our experiments, conducted on various entity disambiguation benchmarks, demonstrate the model’s strong and robust performance. Furthermore, we integrate this approach into the retrieval/reader EL framework and observe improvements on the GERBIL benchmark compared with previous SoTA. We also propose entity retrieval-augmented large language models (LLMs) for EL. Results show that compared to FusionED, LLMs generally underperform while they demonstrate strong improvements compared to SoTA over some datasets.

6 Limitations and Ethical Considerations

The scope of our ED and EL models are limited to traditional Wikipedia and News datasets. We have not investigated its effectiveness in diverse domains such as biomedical research, e-commerce, and product catalogs. Furthermore, this paper focuses exclusively on the English corpus, and exploring the potential of our model in amultilingual setting would be an interesting expansion for future research. This includes investigating the advantages of projecting entity linking concepts from one language to another and employing multilingual representation learning to enhance our base model. While our retrieval-augmented LLMs exhibit notable performance improvements for certain datasets in EL, they underperform compared to the other approaches. Investigating how to enhance the performance of LLMs using different prompt further is an interesting direction for exploration.

Our models are trained using datasets comprised of existing textual collections sourced from Wikipedia and News. Recent studies have brought attention to potential societal biases ingrained in established corpora. We acknowledge the potential risk that our EL models may inherit such biases.

References

Nicholas Botzer, Yifan Ding, and Tim Weninger. 2021. Reddit entity linking dataset. *Information Processing & Management*, 58(3):102479.

Samuel Broscheit. 2019. Investigating entity knowledge in bert with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685.

Samuel Broscheit. 2020. Investigating entity knowledge in bert with simple neural end-to-end entity linking. *arXiv preprint arXiv:2003.05473*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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)*, pages 708–716.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Highly parallel autoregressive entity linking with discriminative correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7662–7669.

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.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke Van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. 2020. Empirical evaluation of pretraining strategies for supervised entity linking. *arXiv preprint arXiv:2005.14253*.

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

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

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

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554.

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.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. *arXiv preprint arXiv:1808.07699*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart:

Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Belinda Z Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. *arXiv preprint arXiv:2010.02413*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pedro Henrique Martins, Zita Marinho, and André FT Martins. 2019. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196.

Marcel Milich and Alan Akbik. 2023. ZELDA: A comprehensive benchmark for supervised entity disambiguation. In *EACL 2023, The 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. 2015. Open knowledge extraction challenge. In *Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31-June 4, 2015, Revised Selected Papers*, pages 3–15. Springer.

Laurel Orr, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling, and Christopher Re. 2020. Bootleg: Chasing the tail with self-supervised named entity disambiguation. *arXiv preprint arXiv:2010.10363*.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544.

Vera Provatorova, Samarth Bhargav, Svitlana Vakulenko, and Evangelos Kanoulas. 2021. Robustness evaluation of entity disambiguation using prior probes: the case of entity overshadowing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10501–10510, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. 2014. N^3-a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *LREC*, pages 3529–3533.

Hassan S. Shavarani and Anoop Sarkar. 2023. SpEL: Structured prediction for entity linking. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Nadine Steinmetz and Harald Sack. 2013. Semantic multimedia information retrieval based on contextual descriptions. In *The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings 10*, pages 382–396. Springer.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, et al. 2015. Gerbil: general entity annotator benchmarking framework. In *Proceedings of the 24th international conference on World Wide Web*, pages 1133–1143.

Johannes M Van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. 2022. Global entity disambiguation with bert. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3264–3271.

Yi Yang, Ozan Irsoy, and Kazi Shefaet Rahman. 2018. Collective entity disambiguation with structured gradient tree boosting. *arXiv preprint arXiv:1802.10229*.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2022. EntQA: Entity linking as question answering. In *International Conference on Learning Representations*.

Wenzheng Zhang and Karl Stratos. 2021. Understanding hard negatives in noise contrastive estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1090–1101.

A Additional experiments on named entity disambiguation benchmark

We also run a small ablation experiment on traditional named entity disambiguation datasets using FLAN-T5-large as base model to compare the corresponding large model. Unlike a standard benchmark, models which test on those datasets typically trained using different corpus and linked to different KB which maybe subset of YAGO (Suchanek et al., 2007) and KILT (Petroni et al., 2021). Reproducing those results might be a challenge due to the incomplete release of their entity vocabulary [1] [2]. And comparison is indirect since training datasets are different and may overlap with some test datasets used in out-of-domain evaluation [3].

We avoid training our model on Wikipedia datasets to prevent test data leakage. Instead, we conduct ablation experiments, training on AIDA and evaluating it in both in-domain AIDA-B and out-domain datasets such as MSNBC, AQUAINT, ACE2004, WNED-CWEB (CWEB), and WNED-WIKI (WIKI) (Gabrilovich et al., 2013; Guo and Barbosa, 2018) to provide a direct comparison.

At the inference, we rely on the same candidate lists provided in (De Cao et al., 2021) [4]. Instead of decoding entity names, we decode the corresponding entity number in the given ordered candidate list.

Table 5 presents a comparison of InKB Micro F1 results between GENER and FUSIONED when trained on the AIDA dataset and evaluated in both in-domain and out-of-domain scenarios. FUSIONED shows much better performance compared to GENER, supporting our claim that our model does not require significant pre-training. It is worth noting that our numbers are not directly comparable with SoTA models, as those models are trained on different corpus.

B Entity linking experiments in GPT-4

Our prompt template is as follows:

```
Given a input passage and a
candidate entity list (each
element in this list is a pair
```

[1] https://github.com/facebookresearch/GENRE/issues/26

[2] https://github.com/facebookresearch/GENRE/issues/72

[3] https://github.com/facebookresearch/GENRE/issues/13

[4] https://github.com/facebookresearch/GENRE/tree/main/examples_genre

	In-domain	Out-of-domain					
Method	AIDA-B	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	AVG
De Cao et al. (2021)	88.6	88.1	77.1	82.3	71.9	71.7	80.0
FusionED	91.7	92.4	82.0	87.1	75.8	78.6	84.6

Table 5: InKB Micro F1 comparison of GENER and FusionED when training in AIDA dataset and evaluate the performance on both in-domain and out-of-domain. The goal of this experiments is to provide a direct comparison.

```
with entity title and entity
description), your task is to
select entities from this list
and link them to mentions which
appear in given passage.  For
each linkage, please output
the entity title and mention,
separated by @#@ on each line.
You can use the truncated
document as context information.
passage:  ...  , entities:  ...
, document:  ...
```

For each passage, we first retrieve the top-100 entity candidates, then feed this passage, entity candidates, and the corresponding truncated document into this template to produce a prompt. Subsequently, we call the GPT-4-16k API to get results. Then we parse results and evaluate those in GERBIL benchmark.

	GPT-4 + retrieval			GPT-4 + retrieval*		
Dataset	P	R	F1	P	R	F1
AIDA-B	52.0	**66.6**	58.4	**53.2**	66.5	**59.1**
MSNBC	32.6	**60.7**	42.4	**32.8**	60.5	**42.5**
Der	29.2	**63.9**	40.1	**30.2**	**63.9**	**41.0**
K50	70.3	**67.8**	**69.0**	**72.0**	59.4	65.1
R128	25.6	**55.6**	35.1	**27.2**	55.2	**36.4**
R500	19.2	**62.8**	29.4	**20.1**	61.0	**30.1**
OKE15	64.1	**53.5**	58.3	**64.6**	53.3	**58.4**
OKE16	60.7	**47.2**	**53.1**	**61.5**	46.5	53.0

Table 6: Breakdown of the score, Precision (P), Recall (R) and F1 for the GPT-4 + retrieval method.

Table 6 presents the results of GPT-4 in the GERBIL benchmark (Usbeck et al., 2015). For GPT-4 + retrieval (zero-shot)*, we additionally filter entities generated by the model using candidate entities obtained from entity retrieval and this improves its precision and slightly improve its performance over all datasets except K50 and OKE16.