Bidirectional End-to-End Learning of Retriever-Reader Paradigm for Entity Linking

Anonymous ACL submission

Abstract

Entity Linking (EL) is a fundamental task for Information Extraction and Knowledge Graphs. The general form of EL (i.e., end-to-end EL) aims to find mentions in the given document and then link the mentions to corresponding entities in a specific knowledge base. Recently, the paradigm of retriever-reader promotes the progress of end-to-end EL, benefiting from the advantages of dense entity retrieval and machine reading comprehension. However, the existing studies only train the retriever and the reader separately in a pipeline manner, thus ignoring the benefit that the interactions between the retriever and the reader can bring to the task. To advance the retriever-reader paradigm to perform more effectively on end-to-end EL, we propose **BEER**^{2}, a **B**idirectional **E**nd-to-**E**nd training framework for **R**etriever and **R**eader. Through our designed bidirectional end-to-end training, **BEER**^{2} guides the retriever and the reader to learn from each other, make progress together, and ultimately improve EL performance. Extensive experiments on benchmarks of multiple domains demonstrate the effectiveness of our proposed **BEER**^{2}.

1 Introduction

End-to-End Entity Linking (EL) (Shen et al., 2015; Kolitsas et al., 2018; Chen et al., 2020) which is the general form of the EL task, aims to extract mentions from a given text and link the mentions to specific entities in a given knowledge base. Due to its ability to automatically understand text, entity linking has become an essential task for various NLP tasks (Tan et al., 2023), such as knowledge graphs construction (Clancy et al., 2019), automatic text summarization (Amplayo et al., 2018), and question answering (Ferrucci, 2012).

Early end-to-end EL works (Hoffart et al., 2011; Ling et al., 2015; Luo et al., 2015) mainly divide this task into two subtasks, Mention Detection (MD) and Entity Disambiguation (ED), and study how to exploit the potential relationship between them to improve EL performance. Most previous works conduct MD before ED (Nguyen et al., 2016; Martins et al., 2019), which is an unnatural design because it causes models to predict the span position without entity information. This is also the essential reason for the long-standing dilemma in end-to-end EL, that is, “MD is more difficult than ED” (Zhao et al., 2019; Broscheit, 2019). To overcome this challenge, (Zhang et al., 2022) propose EntQA which consists of a retriever-reader structure to perform ED before MD. As shown in Figure 1, given a text, EntQA first retrieves relevant candidate entities from the knowledge base by dense retrieval, and then its reader is responsible for rejecting wrong candidates and extracting the span position information in the text for correct entities. Benefiting from that dense retrieval effectively reduces the huge search space, EntQA achieves state-of-the-art performance and becomes a strong and advanced baseline of end-to-end EL.

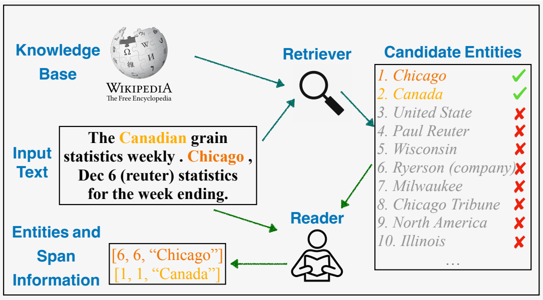


Figure 1: An example of entity linking according to the retriever-reader paradigm.

However, EntQA only trains its retriever and reader separately in a pipeline manner, that is, the training of the reader will be performed after the retriever is fully trained. We argue that this pipelined training cannot enable sufficient interactions between the retriever and the reader. Intuitively, the training signal of the retriever can dynamically affect the training process of the reader, and the results of the reader can also be fed back to the retriever to guide its training. Therefore, it is worth studying how to utilize interactions between the retriever and reader to improve the end-to-end EL performance.

Motivated by the above intuition, we propose the **B**idirectional **E**nd-to-**E**nd learning of **R**etriever-**R**eader (**BEER**^{2}), a more effective training framework that aims to advance the retriever-reader paradigm to perform more perfectly on end-to-end EL. The **BEER**^{2} contains two data flows in opposite directions: (1) Retriever → Reader. The retriever dynamically gets candidate entities and inputs them into the reader, thereby updating the training data of the reader in real time. (2) Reader → Retriever. The reader identifies mentions in the documents and inputs the span position information into the retriever, which in turn allows the retriever to perform more effective span-based retrieval. Through these two bidirectional data flows, we jointly train the retriever and reader in an end-to-end manner and then guide them to learn from each other, make progress together, and finally improve the EL performance. In addition, we believe that the core idea of our proposed **BEER**^{2} is also useful for enhancing retriever-reader likely models in other tasks, such as open-domain question answering.

In summary, our contributions are in three folds:

1. We present the end-to-end **BEER**^{2} framework, which jointly trains the retriever and reader to make them interact and enhance each other.

2. We conduct extensive experiments on benchmarks in two languages (English and Chinese) of multiple domains (including news, speech, and medical domains) and achieve new state-of-the-art end-to-end EL performance.

3. We provide sufficient ablation studies and detailed analyses for better verification of the effectiveness of our proposed method.

2 Methodology

In this section, we introduce the details of **BEER**^{2}, which are illustrated in Figure 2. Our proposed **BEER**^{2} consists of a retriever and a reader, which are trained jointly in an end-to-end manner. From the perspective of the data flow, our approach contains two opposite data flows, namely "Retriever → Reader" and "Reader → Retriever".

2.1 Retriever-Reader Structure

The retriever module aims to retrieve candidate entities that might belong to the input text from the knowledge base, and the purpose of the reader module is to further reduce the candidate set and predict the specific span position information of the finally predicted entities in the text.

2.1.1 Retriever

Let the knowledge base be denoted by $\mathcal{KB} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_N\}$. Given a sentence $\boldsymbol{t}$ of length $T_{\boldsymbol{t}}$, the retriever module is to achieve a subset $\mathcal{E}_{\text{cand}} \subset \mathcal{KB}$ to be the candidate entities of $\boldsymbol{t}$. Specifically, we model the retriever as a dual-encoder (Bromley et al., 1993), which contains a sentence encoder $\mathrm{E_S}$ and an entity encoder $\mathrm{E_E}$. We use $\mathrm{E_S}$ to map the sentence $\boldsymbol{t}$ to a sequences of representations $\boldsymbol{r^t}$:

$$\begin{aligned} \boldsymbol{r^t} &= \mathrm{E_S}(\boldsymbol{t}), \\ &= [r^{\boldsymbol{t}}_{[\text{CLS}]}, r^{\boldsymbol{t}}_1, ..., r^{\boldsymbol{t}}_{T_{\boldsymbol{t}}}], \end{aligned} \tag{1}$$

where [CLS] is the special token representing the beginning of a sequence in the tokenizer of BERT (Devlin et al., 2019). And we use $\mathrm{E_E}$ to get the representations $\boldsymbol{r^{e_i}}$ of the entity $\boldsymbol{e}_i \in \mathcal{KB}$:

$$\begin{aligned} \boldsymbol{r^{e_i}} &= \mathrm{E_E}(f(\boldsymbol{e}_i)), \\ &= [r^{\boldsymbol{e}_i}_{[\text{CLS}]}, r^{\boldsymbol{e}_i}_1, ..., r^{\boldsymbol{e}_i}_{T_{\boldsymbol{e}}}], \end{aligned} \tag{2}$$

where $f(\boldsymbol{e}_i)$ represents the operation of obtaining the description text of $\boldsymbol{e}_i$ from $\mathcal{KB}$. It is worth noting that we uniformly limit the description text length of all entities in $\mathcal{KB}$ to $T_{\boldsymbol{e}}$.

After obtaining the representations of the sentence and entities, we select entities based on their retrieval scores. We use the dot product between vectors as our scoring function. We first use the [CLS] representations of sentences and entities to retrieve the top K entities:

$$\mathcal{E}_{\text{cls}} = \underset{\mathcal{E}' \subset \mathcal{E}, |\mathcal{E}'|=K}{\arg\max} \sum_{\boldsymbol{e}_i \in \mathcal{E}'} r^{\boldsymbol{t}\top}_{[\text{CLS}]} r^{\boldsymbol{e}_i}_{[\text{CLS}]}. \tag{3}$$

In addition, to enhance the interactions between the retriever and the reader, we also utilize the span information predicted by the reader for auxiliary retrieval. Assuming that the reader's prediction for the spans in $\boldsymbol{t}$ is $\boldsymbol{s} = \{s_i\}, 1 \leq s_i \leq T_{\boldsymbol{t}}$, we use it to obtain more accurate span representations than [CLS] representations as follows:

$$r^{\boldsymbol{t}}_{\text{span}} = \text{avg}([r^{\boldsymbol{t}}_{s_i}]), s_i \in \boldsymbol{s}, \tag{4}$$

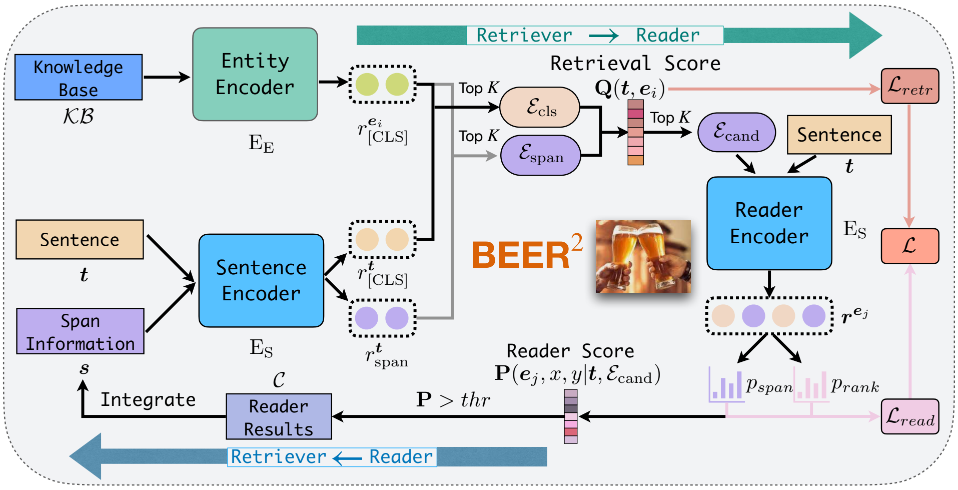


Figure 2: The training process of **BEER**^{2}. Note that encoders of the same color represent that they share parameters.

where $\mathrm{avg}(\cdot)$ is the mean pooling operation. Then we use the span representations to retrieve the top K entities again:

$$\mathcal{E}_{\mathrm{span}} = \underset{\mathcal{E}' \subset \mathcal{E}, |\mathcal{E}'| = K}{\arg\max} \sum_{\boldsymbol{e}_i \in \mathcal{E}'} r^{\boldsymbol{t}\top}_{\mathrm{span}} r^{\boldsymbol{e}_i}_{[\mathrm{CLS}]}. \tag{5}$$

Considering that $\mathcal{E}_{\mathrm{cls}}$ and $\mathcal{E}_{\mathrm{span}}$ may have duplicate entities, we finally take the top K entities of the set of $\{\mathcal{E}_{\mathrm{cls}} \cup \mathcal{E}_{\mathrm{span}}\}$ as the final retrieval result, i.e., $\mathcal{E}_{\mathrm{cand}}$, which will be sent to the reader as input.

2.1.2 Reader

To enable end-to-end training, we make the reader encoder and the retriever's sentence encoder share parameters. Therefore, we denote the reader encoder as $\mathrm{E_S}$ for the convenience of understanding. Given the output of the retriever (i.e., $\mathcal{E}_{\mathrm{cand}}$) and the input sentence $\boldsymbol{t}$, for each candidate entity $\boldsymbol{e}_j \in \mathcal{E}_{\mathrm{cand}}$, we get its joint representation with $\boldsymbol{t}$:

$$\begin{aligned} \boldsymbol{r}^{\boldsymbol{e}_j} &= \mathrm{E_S}(\boldsymbol{t} \oplus f(\boldsymbol{e}_j)), \\ &= [r^{\boldsymbol{e}_j}_{[\mathrm{CLS}]}, r^{\boldsymbol{e}_j}_1, ..., r^{\boldsymbol{e}_j}_{T_{\boldsymbol{t}}}, r^{\boldsymbol{e}_j}_{[\mathrm{SEP}]}, ..., r^{\boldsymbol{e}_j}_{T_{\boldsymbol{t}}+T_e}]. \end{aligned} \tag{6}$$

Based on the joint representation, according to the mechanism proposed in EntQA, we compute the probability of span $(x, y), 1 \leq x, y \leq T_{\boldsymbol{t}}$ and the ranking probability of $\boldsymbol{e}_j, 1 \leq j \leq K$:

$$p_1(x|\boldsymbol{t}, \boldsymbol{e}_j) = \mathrm{softmax}(\boldsymbol{W}_1 \boldsymbol{r}^{\boldsymbol{e}_j})[x], \tag{7}$$

$$p_2(y|\boldsymbol{t}, \boldsymbol{e}_j) = \mathrm{softmax}(\boldsymbol{W}_2 \boldsymbol{r}^{\boldsymbol{e}_j})[y], \tag{8}$$

$$p_{span}(x, y|\boldsymbol{t}, \boldsymbol{e}_j) = p_1(x|\boldsymbol{t}, \boldsymbol{e}_j) \times p_2(y|\boldsymbol{t}, \boldsymbol{e}_j), \tag{9}$$

$$p_{rank}(\boldsymbol{e}_j|\boldsymbol{t}, \mathcal{E}_{\mathrm{cand}}) = \frac{\exp\left(\boldsymbol{W}_3^\top r_1^{\boldsymbol{e}_j}\right)}{\sum_{j'=1}^{K} \exp\left(\boldsymbol{W}_3^\top r_1^{\boldsymbol{e}_{j'}}\right)}, \tag{10}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3$ are trainable parameters. Furthermore, for a combination of span and entity, its reader score is computed as:

$$\begin{aligned} \mathbf{P}(\boldsymbol{e}_j, x, y|\boldsymbol{t}, \mathcal{E}_{\mathrm{cand}}) = p_{span}(x, y|\boldsymbol{t}, \boldsymbol{e}_j) \\ \times p_{rank}(\boldsymbol{e}_j|\boldsymbol{t}, \mathcal{E}_{\mathrm{cand}}). \end{aligned} \tag{11}$$

Based on the scores of all possible combinations, we select the combinations that satisfy "$\mathbf{P} > thr$" as the final reader prediction results. The thr is the threshold we choose empirically. We denote the N reader results as $\mathcal{C} = \{(\boldsymbol{e}_l, x_l, y_l)\}, 1 \leq l \leq N$. We integrate the span position (i.e., $\{(x_l, y_l)\}$) into $\boldsymbol{s} = \{s_i\}$ and send $\boldsymbol{s}$ to the retriever.

2.2 Bidirectional Data Flows

The key innovation of BEER^2 compared to EntQA is to use two bidirectional data flows so that the retriever and the reader are trained in an end-to-end fashion. Their enhanced interaction allows them to obtain positive affect from each other and improve performance.

2.2.1 Retriever → Reader

In the framework of **BEER**^{2}, the retriever sends its retrieval results $\mathcal{E}_{\mathrm{cand}}$ to the reader after each time it completes the retrieval. In fact, in EntQA, the input of the reader is also the output of the retriever. However, the pipeline training method of EntQA determines that only when the retriever's training ends, the inference result of the retriever module can be used as the training input of the reader. We think this is a kind of data interaction that is relatively hard and dull, and the reader cannot perceive the signal change of the retriever training process, because the reader can only receive the inference result of the retriever. Therefore, unlike EntQA, **BEER**^{2} dynamically sends the retriever results to the reader during the training process, which allows the reader to learn the experience gained by the retriever when they are training. Additionally, the end-to-end property of the **BEER**^{2} framework enables the retriever and reader to receive gradient propagation at the same time during training. If the retriever selects the wrong candidate entity, this will cause the reader to be affected as well, and the signal of the reader will also be directly fed back to the retriever so that it can correct the error in time. Particularly, we believe that if the signal of the ranking probability p_{rank} calculated in the reader is propagated to the retriever, it will be very helpful for the optimization of the retriever, because the ranking probability is to score and sort the results of the retriever, so this can obviously be regarded as the training rewards of the retriever.

2.2.2 Reader → Retriever

From Equations 4 and 5, we know the key information connecting the bridge from the reader to the retriever is the span position information $\boldsymbol{s}$. According to the span prediction results of the reader, our retriever accurately extracts the span representations of the input sentence, and then performs auxiliary entity retrieval. There are two main motivations for us to use the span prediction results to assist retrieval: (1) The [CLS] representation of a sentence can reflect the semantics of the entire sentence, but it is not sufficient for the representation of the entities in the sentence. Hence, we think only using [CLS] representations is not optimal for entity-centric tasks (Li et al., 2022; Zaporojets et al., 2022; Wang et al., 2022) like EL. The span representations can not only make the retriever perceive the location of the mentions in the sentence, but also improve the retrieval diversity. (2) Also benefiting from our parameter sharing setting, during training, if the reader makes a wrong span prediction, then this will cause the retriever to fail to obtain an accurate span representation and make a wrong entity selection, and the retriever's loss gradient will be passed back to the reader so that it can learn and get progress.

Besides, in practice, to allow the retriever to have span information input at the beginning of training, we arrange a process similar to the model's warm-up before starting formal training. This process will sequentially pre-train the retriever and reader with a small number of epochs, so as to obtain the span information which will be used for the initial input of the retriever in the formal end-to-end training of **BEER**^{2}. It is worth noting that this warm-up-like process does not cause an unfair comparison between our method and EntQA, which we will empirically prove in Section 3.5.3.

2.3 Overall Training Objective

Given a training sentence $\boldsymbol{t}$ and the knowledge base $\mathcal{KB}$, based on the Equations 3 and 5, for an entity $\boldsymbol{e}_i \in \mathcal{KB}$, the retriever's score is defined as:

$$\mathbf{Q}(\boldsymbol{t}, \boldsymbol{e}_i) = r^{\boldsymbol{t}\top}_{\text{[CLS]}} r^{\boldsymbol{e}_i}_{\text{[CLS]}} + r^{\boldsymbol{t}\top}_{\text{span}} r^{\boldsymbol{e}_i}_{\text{[CLS]}}. \tag{12}$$

For the training sentence $\boldsymbol{t}$, we have its gold entity set $\mathcal{G} \subset \mathcal{KB}$, and we can achieve its negative entity set $\mathcal{N} \subset \mathcal{KB} \setminus \mathcal{G}$. Then we train the retriever by Noise Contrastive Estimation objective (Gutmann and Hyvärinen, 2010) which is defined as:

$$\begin{gathered}\mathcal{L}_{retr} = \max \sum_{\boldsymbol{g} \in \mathcal{G}} \log \\ \left(\frac{\exp\left(\mathbf{Q}(\boldsymbol{t}, \boldsymbol{g})\right)}{\exp\left(\mathbf{Q}(\boldsymbol{t}, \boldsymbol{g})\right) + \sum_{\boldsymbol{n} \in \mathbf{N}} \exp\left(\mathbf{Q}\left(\boldsymbol{t}, \boldsymbol{n}\right)\right)} \right).\end{gathered} \tag{13}$$

As for the training of the reader, we directly optimize it to maximize the span probability (i.e., p_{span}) and ranking probability (i.e., p_{rank}):

$$\begin{aligned}\mathcal{L}_{read} = \max \sum_{j=1}^{K} \sum_{(x,y)} (&\log p_{span}(x, y | \boldsymbol{t}, \boldsymbol{e}_j) + \\ &\log p_{rank}(\boldsymbol{e}_j | \boldsymbol{t}, \mathcal{E}_{\text{cand}})).\end{aligned} \tag{14}$$

It is worth noting that, during the training process, the combination of the (x, y) is the gold mention spans of every candidate entity $\boldsymbol{e}_j \subset \mathcal{E}_{\text{cand}}$.

Finally, we train the retriever objective $\mathcal{L}_{retr}$ and reader objective $\mathcal{L}_{read}$ simultaneously. The overall end-to-end objective of **BEER**^{2} is defined as:

$$\mathcal{L} = \mathcal{L}_{retr} + \mathcal{L}_{read}. \tag{15}$$

3 Experiments

3.1 Datasets

To evaluate **BEER**^{2} comprehensively, we select EL datasets from multiple domains. Particularly, it is well known that the medical domain is very special due to its professional nature, so medical EL has long been regarded as an independent task (Mondal et al., 2019). But we run **BEER**^{2} on both the medical domain and other generic domains. Additionally, our work is the first to use multilingual benchmarks, including English and Chinese. The dataset details are shown in Appendix A.

- **News:** The AIDA-CoNLL dataset (Hoffart et al., 2011) contains 1,393 English news articles from Reuters. Its entities are identified by YAGO2 entity name and Wikipedia URL.
- **Medical:** The BC5CDR dataset (Li et al., 2016) contains 1,500 medical abstracts that are annotated with MeSH ontology.
- **Speech:** The NLPCC2022 dataset (Song et al., 2022) is the benchmark for the public competition of NLPCC2022. It consists of 1,936 TED talks converted from raw audio.
- **Short-Text:** The CCKS2020 dataset [1] is provided by CCKS2020 Chinese short text EL task. Its corpus comes from various domains, such as movies, TV, novels, etc.

3.2 Baseline Methods

To reflect the competitiveness of **BEER^2**, we select several advanced strong baselines: **End2End EL** (Kolitsas et al., 2018) uses a Bi-LSTM (Hochreiter and Schmidhuber, 1997) to encode embeddings and links mention to entities based on local and global scores. **Joint NER EL** (Martins et al., 2019) propose to jointly learn the NER task and EL task to make them benefit from each other. **REL** (van Hulst et al., 2020) is a widely used open-source toolkit for entity linking. It is an ensemble of multiple state-of-the-art NLP methods and packages. **GENRE** (Cao et al., 2021) models the EL task as a seq2seq problem and automatically generates unique entity identifiers of the input guiding text. **ReFinED** (Ayoola et al., 2022) is an efficient zero-shot-capable method for end-to-end EL. It introduces the fine-grained entity typing task to improve the performance of EL. **EntQA** [2] (Zhang et al., 2022) decomposes the end-to-end EL task into two subproblems, namely entity retrieval and question answering. EntQA is the previous state-of-the-art method on the AIDA-CoNLL dataset. **DNorm** (Leaman et al., 2013) utilizes the TF-IDF to learn a bilinear mapping function for ED of the medical EL task. **ID-CNN** (Strubell et al., 2017) is a deep learning-based method that uses a CNN network to do the medical NER task. **E2EMERN** (Zhou et al., 2021) is an end-to-end progressive multi-task learning framework for the medical EL task. It achieves the previous state-of-the-art results on the BC5CDR dataset. **KENER** (Huang et al., 2022) focuses on incorporating proper knowledge in the MD subtask to improve the overall performance of linking. It was the best system in the competition of NLPCC2022.

3.3 Evaluation Metric

To ensure the fairness of the comparison between our method and baselines, the metrics we use to report our main results are the widely used InKB Micro Precision, Recall, and F1 score. Specifically, for the end-to-end EL task, a mention is considered to be correct only when its span position is extracted correctly and the corresponding entity id in the knowledge base is predicted correctly. It should be emphasized that when evaluating EL performance, the F1 score is considered as the primary metric. The F1 score is the harmonic mean of Precision and Recall, meaning it takes both of them into account when calculating overall performance and effectively balances the trade-off between Precision and Recall. A high F1 score indicates that model performs well in terms of correctly identifying and linking entities while minimizing false positives and false negatives. In addition, because both EntQA and **BEER^2** are models of the retriever-reader paradigm, we report the retriever's Recall@K to reflect the retrieval performance.

Other details of experimental implementation are presented in Appendix B.

3.4 Experimental Results

From Table 1, we can see that:

1. **BEER^2** outperforms the previous state-of-the-art models on all datasets. Specifically, **BEER^2** exceeds EntQA by 1.2% F1 on AIDA-CoNLL, exceeds E2EMERN by 0.8% F1 on BC5CDR, exceeds KENER by 2.2% F1 on NLPCC2022, exceeds EntQA by 1.9% F1 on CCKS2020. The strong results of **BEER^2** demonstrate the effectiveness of our proposed bidirectional end-to-end learning of the retriever-reader paradigm for EL.

2. Compared with EntQA, the improvements of **BEER^2** are significant. For the domains of medicine and speech, in which the performance of EntQA is not the best, **BEER^2** outperforms it by 2.6% F1 and 2.8% F1 respectively and becomes the best model in these two domains. This indicates that **BEER^2** has better domain adaptation ability.

[1] http://biendata.xyz/competition/ccks_2020_el

[2] https://github.com/WenzhengZhang/EntQA

Language	Domain	Dataset	Method	InKB Micro Precision	 Recall	 F1 Score
English	**News**	AIDA-CoNLL	End2End EL (Kolitsas et al., 2018)	80.9	84.0	82.4
			Joint NER EL (Martins et al., 2019)	81.1	82.8	81.9
			REL (van Hulst et al., 2020)	79.5	81.5	80.5
			GENRE (Cao et al., 2021)	81.7	85.8	83.7
			ReFinED (Ayoola et al., 2022)	81.8	86.3	84.0
			EntQA (Zhang et al., 2022)	84.6	87.0	85.8
			BEER^2 (Ours)	**86.9**$^{\uparrow}$	**87.2**$^{\uparrow}$	**87.0**$^{\uparrow}$
English	**Medical**	BC5CDR	DNorm (Leaman et al., 2013)	82.7	78.7	80.7
			IDCNN (Strubell et al., 2017)	82.0	80.3	81.1
			E2EMERN (Zhou et al., 2021)	82.5	**82.1**	82.3
			EntQA (Zhang et al., 2022)	81.8	81.2	81.5
			BEER^2 (Ours)	**86.0**$^{\uparrow}$	80.3	**83.1**$^{\uparrow}$
English	**Speech**	NLPCC2022	KENER (Huang et al., 2022)	-	-	74.6
			EntQA (Zhang et al., 2022)	76.0	72.1	74.0
			BEER^2 (Ours)	**76.1**$^{\uparrow}$	**77.5**$^{\uparrow}$	**76.8**$^{\uparrow}$
Chinese	**Short-Text**	CCKS2020	EntQA (Zhang et al., 2022)	75.5	70.7	73.0
			BEER^2 (Ours)	**77.1**$^{\uparrow}$	**72.8**$^{\uparrow}$	**74.9**$^{\uparrow}$

Table 1: The performance of **BEER**^{2} and all baselines. We underline the previous state-of-the-art results. Note that the results of E2EMERN are obtained by running its officially trained model under our metrics.

3. For the speech domain, **BEER**^{2} outperforms KENER, which reflects the competitiveness of **BEER**^{2}. KENER is a competition system that includes an ensemble method. Without this trick that is useful in improving performance, **BEER**^{2} is still better than KENER. Additionally, the better performance compared with EntQA on the Chinese benchmark also reflects the language robustness of **BEER**^{2}.

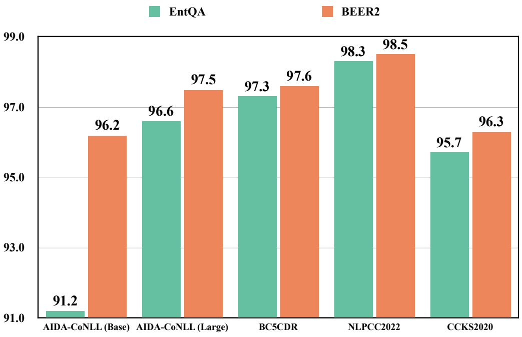


Figure 3: The retrieval performance (Recall@K) of EntQA and **BEER**^{2}. Particularly, on AIDA-CoNLL, we use BERT-Base/Large to initialize respectively.

Method	Recall@K	F1
EntQA (Base)	91.2	79.1
BEER^2 (Base, Retriever → Reader)	93.1	80.3
BEER^2 (Base, Reader → Retriever)	95.8	81.4
BEER^2 (Base, Retriever ↔ Reader)	96.2	81.7
EntQA (Large)	96.6	85.8
BEER^2 (Large, Retriever → Reader)	96.8	86.0
BEER^2 (Large, Reader → Retriever)	97.2	86.3
BEER^2 (Large, Retriever ↔ Reader)	97.5	87.0
BEER^2	**97.5**	**87.0**

Table 2: The retriever and reader performance of the variants of **BEER**^{2} on AIDA-CoNLL.

3.5 Analysis and Discussion

3.5.1 The Retrieval Performance

Table 1 reports the reader performance as our main results. As a study of the retriever-reader structure, it is necessary to analyze the retrieval performance. From Figure 3, we see that **BEER**^{2} always has better retrieval performance than EntQA, which verifies the advantage of our designed retriever module. When Bert-Base is used as the backbone, the result of EntQA is relatively low, which leaves more space for **BEER**^{2}. Therefore, our method indeed improves significantly. Furthermore, we also find that the improvements on AIDA-CoNLL and CCKS2020 are greater than that on BC5CDR and NLPCC2022. For this phenomenon, we suspect that it is because the knowledge bases of AIDA-CoNLL and CCKS2020 have more entities than BC5CDR and NLPCC2022. A larger number of entities pose a greater challenge to the retriever, thus, better performance on larger knowledge bases reflects that our retriever is better than EntQA's.

3.5.2 Effects of Two Data Flows

Our technical contribution is that we design an end-to-end training mechanism, which includes two

Method	Pre	Rec	F1
EntQA (4 epochs)	84.6	87.0	85.8
EntQA (20 epochs)	85.4	86.4	85.9
BEER^2 (1 epoch + 10 epochs)	86.6	87.4	87.0
BEER^2 (5 epochs + 10 epochs)	86.9	87.2	87.0

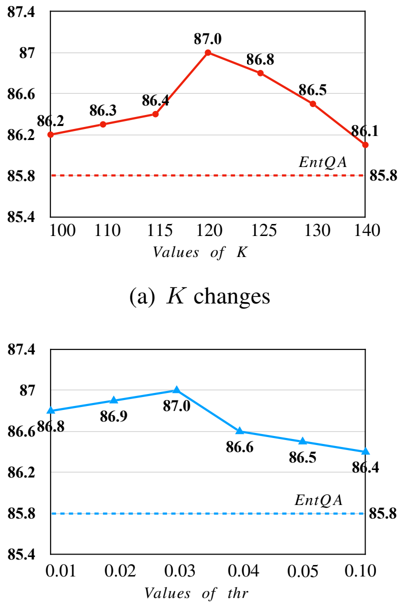
Table 3: The performance when training models with different epochs on AIDA-CoNLL.

bidirectional data flows as bridges connecting the retriever and reader modules. Therefore, we further conduct ablation studies on these two data flows.

From Table 2, we see that each of the data flows we design individually brings considerable improvements. As described in Sections 2.2.1 and 2.2.2, while dynamically inputting the candidate entities from the retriever effectively helps the reader's training, thanks to the end-to-end training, the retriever itself is also further optimized. This view can be seen from the results that **BEER**^{2} (Retriever → Reader) is better than EntQA on both Recall@K and F1. Similarly, from the comparison of the results of **BEER**^{2} (Reader → Retriever) and EntQA, it can be known that the span information sent from the reader to the retriever not only effectively assists the work of the retriever, but also makes its own progress in the prediction of the span position. Besides, the results of **BEER**^{2} (Retriever ↔ Reader) show that these two data flows cooperate well in the framework of **BEER**^{2}, resulting in better performance than they obtain alone.

3.5.3 Effects of Training Epochs

To verify that the warm-up-like pre-training process before the formal training starts does not cause unfairness in model comparison, we train EntQA and **BEER**^{2} for different epochs. Note that **BEER**^{2} (1 epoch + 10 epochs) of Table 3 means that we pre-train the retriever and reader for 1 epoch and then formally train them for 10 epochs under the end-to-end setting. From Table 3, even EntQA is trained for 20 epochs, which means that the number of training epochs is more than that of **BEER**^{2}, its performance is only slightly improved compared to EntQA (4 epochs). This shows that simply increasing the number of training epochs does not bring substantial performance improvements for EntQA. Moreover, the slight difference in results between **BEER**^{2} (1 epoch + 10 epochs) and **BEER**^{2} (5 epochs + 10 epochs) also indicates that the number of epochs of the warm-up-like process is not critical to the training of **BEER**^{2}. The role of this process is only to provide initial span information for our designed retriever. The great advantage of **BEER**^{2} (5 epochs + 10 epochs) compared to EntQA (20 epochs) also empirically proves what we mentioned in Section 2.2.2, that is, the warm-up-like pre-training process will not cause unfairness in the comparison between EntQA and **BEER**^{2}.



(a) K changes

(b) thr changes

Figure 4: The F1 results of **BEER**^{2} on AIDA-CoNLL.

3.5.4 Parameter Studies of K and thr

Figure 4(a) presents the performance change of **BEER**^{2} as choosing different values of K. We see that as the value of K increases, the performance of **BEER**^{2} shows a trend of first increasing and then decreasing. This phenomenon is in line with our intuition because K represents the number of candidate entities sent to the reader. If K is too large, it produces more noise entities, thus damaging the reader's performance. However, choosing an excessively large K value itself will not bring much gain to **BEER**^{2}, and will even greatly increase the time spent by retrieving entities. Therefore, choosing an appropriate value of K can obtain competitive performance, after all, **BEER**^{2} performs better than EntQA at all K in Figure 4(a).

In Section 2.1.2, we design to automatically filter predicted combinations of span and entity in the reader. As a key parameter, we carry out the parameter study to verify the insensitivity of **BEER**^{2} to thr. From Figure 4(b), we see that the performance of **BEER**^{2} is not very sensitive to the spe-

Input 1:	pakistan, who arrive next week, are the third team in the triangular world series
Gold:	[0, 0, “Pakistan national cricket team”], [12, 13, “World Series Cricket”]
EntQA:	[5, 5, “Pakistan national cricket team”]
BEER^2:	[5, 5, “Pakistan national cricket team”], [12, 13, “World Series Cricket”]
Input 2:	were not optimistic of a peaceful festive season in kwazulu-natal
Gold:	[11, 15, “KwaZulu-Natal”]
EntQA:	[11, 15, “KwaZulu-Natal”], [11, 15, “KwaZulu”]
BEER^2:	[11, 15, “KwaZulu-Natal”]

Table 4: Examples of EntQA and **BEER**^{2}. We mark the span of mention/golden entity/wrong results. For the EL task, the golden information includes the starting and ending positions of the span and the specific entity.

cific values when thr is within a reasonable range. As thr changes, the F1 score fluctuates slightly in the range greater than 85.0. Therefore, the performance of **BEER**^{2} is robust to the choice of thr. And it can be seen that **BEER**^{2} always outperforms EntQA with the change of thr value.

3.6 Case Study

Table 4 illustrates the comparisons between the cases of EntQA and **BEER**^{2}. In the first case, EntQA does not recognize that “world series” in the sentence is a mention of an entity, while **BEER**^{2} does. We think this is because the retrieval results of **BEER**^{2} are more diverse than that of EntQA because we leverage two kinds of representations for candidate retrieval. In addition, more interestingly, we find that **BEER**^{2} is better than EntQA when dealing with nested entities, as shown in the second example. Because EntQA only uses the overall sentence representation of the [CLS] for retrieval, it cannot perceive the specific position of the span, which leads the model to think that there may be two mentions in the sentence, namely “kwazulu” and “kwazulu-natal”. But **BEER**^{2} knows the specific location of the span when retrieving, that is, it knows that [11, 15] is a span, so it can avoid selecting “kwazulu”. Therefore, the second example reflects the importance of the span information predicted by the reader for the retriever.

4 Related Work

End-to-End Entity Linking (EL) is the general form of EL (Shen et al., 2023; Zhang-li et al., 2022; Joko and Hasibi, 2022). Early works divide the end-to-end EL task into two subtasks, namely Mention Detection (MD)/NER and Entity Disambiguation (ED) (Sil and Yates, 2013), and study the joint learning of these two subtasks to improve EL performance (Luo et al., 2015; Nguyen et al., 2016; Martins et al., 2019). (Kolitsas et al., 2018) develop the first neural end-to-end EL system that considers all potential mentions and calculates contextual similarity scores of candidate entities. Recently, researchers have become enthusiastic about using paradigms of other tasks to improve end-to-end EL (Wu et al., 2020; De Cao et al., 2021; Lai et al., 2022; Cho et al., 2022; Ran et al., 2023).

GENRE (Cao et al., 2021) is an autoregressive model for end-to-end EL. It retrieves entities by generating entity names in an autoregressive mechanism (Dong et al., 2021). ReFinED (Ayoola et al., 2022) separately address the EL task into three subtasks, namely MD, fine-grained entity typing, and ED, thereby enhancing EL with the help of fine-grained entity categories and descriptions. Considering the long-term dilemma of previous works performing MD before ED, that is these methods require models to accurately extract mentions without entity information, EntQA (Zhang et al., 2022) of the retriever-reader paradigm is proposed to solve ED before MD by the way of inverted Open-Domain Question Answering. Thanks to its more natural design, the success of EntQA indicates the advantages of the retriever-reader structure for end-to-end EL. Our work aims to propose a novel EL model in which the retriever and reader are more interactive to facilitate the advancement of the retriever-reader paradigm on end-to-end EL.

5 Conclusion

In this paper, we introduce to promote EL by facilitating the advancement of the retriever-reader paradigm. We design **BEER**^{2}, a bidirectional end-to-end learning framework that enables sufficient retriever-reader interaction. Extensive experiments and analyses show the effectiveness of **BEER**^{2}. In the future, we think it is a promising direction to apply our idea of enhancing the retriever-reader interaction to other related tasks. Besides, our practice of using span representations to assist retrieval is a valuable exploration for dense entity retrieval.

Limitations

As academic verification experiments, we do not consider the running efficiency of our proposed methods. Particularly, to enable the retriever to get the reader's prediction results as input at the beginning of the end-to-end training, we arrange a warm-up-like pre-training process before the end-to-end training begins to obtain the initial span information, as described in Section 2.2.2. Although in Section 3.5.3 we have empirically demonstrated that this process does not cause an unfair comparison of the performance of **BEER**^{2} and baseline models, we must also realize that such a process will make the total training time longer to a certain extent. According to the results in Table 3, it can be known that the epoch number of this warm-up-like process has a slight impact on **BEER**^{2}. Therefore, we suggest that to obtain a balance between efficiency and performance in practice, we can choose a small number of epochs for pre-training.

Besides, another reason why we share the parameters of the retriever's sentence encoder and the reader encoder is to avoid excessive parameters of **BEER**^{2}. The EL task generally occupies a large amount of GPU memory because the number of entities in the knowledge base is large. Therefore, one risk of using different encoders for the retriever and the reader is that it will cause the model training to require too much GPU memory, although this may lead to better performance. The setting of **BEER**^{2}'s shared parameters can effectively reduce its demand for GPU memory. Under our experimental setup, for large-scale encoders, the main GPU requires a maximum of 50G of memory when parameters are shared, while the main GPU requires a maximum of about 75G of memory when parameters are not shared.

Our experiments have proven that the warm-up-like pre-training process before the start of end-to-end training will not bring additional performance gains, this pre-training process with a very small number of epochs has no effect on the model, which suggests that our end-to-end training is essentially equivalent to training from scratch. However, we also admit that such a design is a trade-off considering the convenience of code implementation and engineering development. In the future, we will further study how to make **BEER**^{2} get rid of the dependence on the warm-up-like process, so that our **BEER**^{2} becomes more elegant and efficient. Specifically, we will design how to obtain the initial span information more simply and directly before the end-to-end training starts, without the need for the warm-up-like pre-training process. Moreover, we believe that it is a very interesting and worthwhile research direction to apply our idea of making progress together by learning from each other between retriever and reader to other tasks applicable to the retriever-reader paradigm.

Ethics Considerations

All the datasets, baseline models, and metrics involved in the paper are publicly available. We have cited the corresponding authors or projects of them, and confirm that they are consistent with their intended use.

References

Reinald Kim Amplayo, Seonjae Lim, and Seung-won Hwang. 2018. Entity commonsense representation for neural abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 697–707, New Orleans, Louisiana. Association for Computational Linguistics.

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, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a siamese time delay neural network. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 737–744. Morgan Kaufmann.

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, Hong Kong, China. Association for Computational Linguistics.

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.

Haotian Chen, Xi Li, Andrej Zukov Gregoric, and Sahil Wadhwa. 2020. Contextualized end-to-end neural entity linking. In *Proceedings of the 1st Conference* *of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 637–642. Association for Computational Linguistics.

Young-Min Cho, Li Zhang, and Chris Callison-Burch. 2022. Unsupervised entity linking with guided summarization and multiple-choice selection. In *The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*.

Ryan Clancy, Ihab F. Ilyas, and Jimmy Lin. 2019. Scalable knowledge graph construction from text collections. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 39–46, Hong Kong, China. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese BERT. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3504–3514.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2021. A survey of natural language generation. *CoRR*, abs/2112.11739.

David A. Ferrucci. 2012. Introduction to "this is watson". *IBM J. Res. Dev.*, 56(3):1.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pages 297–304. JMLR.org.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

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, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Shen Huang, Yuchen Zhai, Xinwei Long, Yong Jiang, Xiaobin Wang, Yin Zhang, and Pengjun Xie. 2022. DAMO-NLP at NLPCC-2022 task 2: Knowledge enhanced robust NER for speech entity linking. In *Natural Language Processing and Chinese Computing - 11th CCF International Conference, NLPCC 2022, Guilin, China, September 24-25, 2022, Proceedings, Part II*, volume 13552 of *Lecture Notes in Computer Science*, pages 284–293. Springer.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.

Hideaki Joko and Faegheh Hasibi. 2022. Personal entity, concept, and named entity linking in conversations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 4099–4103. ACM.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Tuan Lai, Heng Ji, and ChengXiang Zhai. 2022. Improving candidate retrieval with entity profile generation for Wikidata entity linking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3696–3711, Dublin, Ireland. Association for Computational Linguistics.

Robert Leaman, Rezarta Islamaj Dogan, and Zhiyong Lu. 2013. Dnorm: disease name normalization with pairwise learning to rank. *Bioinform.*, 29(22):2909–2917.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative V CDR task corpus: a resource for chemical disease relation extraction. *Database J. Biol. Databases Curation*, 2016.

Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022. Contrastive learning with hard negative entities for entity set expansion. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1077–1086. ACM.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal. Association for Computational Linguistics.

Pedro Henrique Martins, Zita Marinho, and André F. T. 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, Florence, Italy. Association for Computational Linguistics.

Ishani Mondal, Sukannya Purkayastha, Sudeshna Sarkar, Pawan Goyal, Jitesh Pillai, Amitava Bhattacharyya, and Mahanandeeshwar Gattu. 2019. Medical entity linking using triplet network. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 95–100, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2016. J-NERD: Joint named entity recognition and disambiguation with rich linguistic features. *Transactions of the Association for Computational Linguistics*, 4:215–229.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 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, Online. Association for Computational Linguistics.

Chenwei Ran, Wei Shen, Jianbo Gao, Yuhan Li, Jianyong Wang, and Yantao Jia. 2023. Learning entity linking features for emerging entities. *IEEE Trans. Knowl. Data Eng.*, 35(7):7088–7102.

Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. 2023. Entity linking meets deep learning: Techniques and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):2556–2578.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2369–2374. ACM.

Ruoyu Song, Sijia Zhang, Xiaoyu Tian, and Yuhang Guo. 2022. Overview of the NLPCC2022 shared task on speech entity linking. In *Natural Language Processing and Chinese Computing - 11th CCF International Conference, NLPCC 2022, Guilin, China, September 24-25, 2022, Proceedings, Part II*, volume 13552 of *Lecture Notes in Computer Science*, pages 294–299. Springer.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Zeqi Tan, Shen Huang, Zixia Jia, Jiong Cai, Yinghui Li, Weiming Lu, Yueting Zhuang, Kewei Tu, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. DAMO-NLP at semeval-2023 task 2: A unified retrieval-augmented system for multilingual named entity recognition. *CoRR*, abs/2305.03688.

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*, SIGIR '20, page 2197–2200, New York, NY, USA. Association for Computing Machinery.

Xinyu Wang, Jiong Cai, Yong Jiang, Pengjun Xie, Kewei Tu, and Wei Lu. 2022. Named entity and relation extraction with multi-modal retrieval. *CoRR*, abs/2212.01612.

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, Online. Association for Computational Linguistics.

Klim Zaporojets, Lucie-Aimée Kaffee, Thomas Demeester, Chris Develder, and I Augenstein. 2022. Tempel: Linking dynamically evolving and newly emerging entities.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2022. Entqa: Entity linking as question answering. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Daniel Zhang-li, Jing Zhang, Jifan Yu, Xiaokang Zhang, Peng Zhang, Jie Tang, and Juanzi Li. 2022. HOSMEL: A hot-swappable modularized entity linking toolkit for Chinese. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 214–223, Dublin, Ireland. Association for Computational Linguistics.

Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. 2019. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 817–824. AAAI Press.

Baohang Zhou, Xiangrui Cai, Ying Zhang, and Xiaojie Yuan. 2021. An end-to-end progressive multi-task learning framework for medical named entity recognition and normalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6214–6224, Online. Association for Computational Linguistics.

A Dataset Details

The dataset statistics are shown in Table 5.

- **News:** The AIDA-CoNLL dataset (Hoffart et al., 2011) contains 1,393 English news articles from Reuters. Its entities are identified by YAGO2 entity name and Wikipedia URL. Following previous works (Cao et al., 2021; Zhang et al., 2022), we split the AIDA-CoNLL dataset into training (946 documents), development (216 documents), and test (231 documents) sets. We use KILT (Petroni et al., 2021) which contains 5,903,530 entities as the given knowledge base.

- **Medical:** The BC5CDR dataset (Li et al., 2016) contains 1,500 medical abstracts that are annotated with MeSH ontology. Same as related works (Zhao et al., 2019; Zhou et al., 2021), we equally divide it into training, development, and test sets. Because the BC5CDR corpus is annotated with MeSH ontology, we also use MeSH which has 2,311 medical concepts as the knowledge base.

- **Speech:** The NLPCC2022 dataset (Song et al., 2022) is the benchmark for the public competition of the conference of NLPCC2022. It consists of 1,936 TED talks converted from raw audio. We manually and randomly split the full dataset into 4:1 for training and testing. Besides, the competition organizers officially provide a knowledge base constructed based on Wikidata. This knowledge base contains 118,795 entities.

- **Short-Text:** The CCKS2020 dataset [3] is provided by CCKS2020 Chinese short text EL task. Its corpus comes from various domains, such as movies, TV, novels, etc. The training data includes 70,000 sentences and the validation/test includes 10,000 sentences. The given knowledge base is from the Baidu Baike and includes approximately 360,000 entities.

B Implementation Details

Our codes are implemented using Pytorch (Paszke et al., 2019). The architectures of the encoders (i.e., retrieval dual-encoder and reader encoder) we use are BERT$_{\text{LARGE}}$-like models. For different domains, we use different backbone parameters to initialize the base encoders. For the news and speech domains, we initialize encoders with pre-trained BLINK (Wu et al., 2020). For the initial parameter of encoders of the medical domain and Chinese language, we select Biobert-Large (Lee et al., 2020) and Chinese-Roberta-Wwm-Ext (Cui et al., 2021). We train **BEER**^{2} with the Adam (Kingma and Ba, 2015) optimizer for 10 epochs. Our model is trained with linear decay and learning rate warming up. The initial retriever learning rate is set to 2e-6 and the initial reader learning rate is set to 1e-5. The training batch size is set to 8 and the evaluation batch size is set to 32. The number of retrieved candidate entities K is set to 120 by default. We choose the default threshold parameter thr as 0.03. We break up each document into sentences of $T_t = 32$ and pad the description text of entities in the knowledge base to $T_t = 128$. We use Faiss [4] (Johnson et al., 2021) for fast entity retrieval.

Dataset	#Train	#Dev	#Test	#KB
AIDA-CoNLL	946	216	231	5,903,530
BC5CDR	500	500	500	2,311
NLPCC2022	1,549	387	387	118,795
CCKS2020	70,000	10,000	10,000	360,000

Table 5: Statistics of the datasets that we use. #Train/#Dev/#Test represents the number of documents in Training/Development/Test sets respectively, and #KB represents the number of entities in the knowledge base used by the corresponding dataset.

[3] http://biendata.xyz/competition/ccks_2020_el

[4] https://github.com/facebookresearch/faiss