

# Effective Few-Shot Named Entity Linking by Meta-Learning

Xiuxing Li<sup>\*§</sup>, Zhenyu Li<sup>\*§</sup>, Zhengyan Zhang<sup>\*</sup>, Ning Liu<sup>†</sup>, Haitao Yuan<sup>\*</sup>, Wei Zhang<sup>†</sup>,  
Zhiyuan Liu<sup>\*</sup>, Jianyong Wang<sup>\*x</sup>

<sup>\*</sup>Department of Computer Science and Technology, Tsinghua University, Beijing, China  
Email: {lix16, zhengyanzhang14, yht16}@mails.tsinghua.edu.cn, lizhenyuucs@gmail.com,  
liuzy@tsinghua.edu.cn, jianyong@mail.tsinghua.edu.cn

<sup>†</sup>School of Computer Science and Techniques, East China Normal University, Shanghai, China  
Email: zhangwei.thu2011@gmail.com

<sup>‡</sup>School of Software, Shandong University, Jinan, China  
Email: victorliucs@gmail.com

**Abstract**—Entity linking aims to link ambiguous mentions to their corresponding entities in a knowledge base, which is significant and fundamental for various downstream applications, e.g., knowledge base completion, question answering, and information extraction. While great efforts have been devoted to this task, most of these studies follow the assumption that large-scale labeled data is available. However, when the labeled data is insufficient for specific domains due to labor-intensive annotation work, the performance of existing algorithms will suffer an intolerable decline. In this paper, we endeavor to solve the problem of few-shot entity linking, which only requires a minimal amount of in-domain labeled data and is more practical in real situations. Specifically, we firstly propose a novel weak supervision strategy to generate non-trivial synthetic entity-mention pairs based on mention rewriting. Since the quality of the synthetic data has a critical impact on effective model training, we further design a meta-learning mechanism to assign different weights to each synthetic entity-mention pair automatically. Through this way, we can profoundly exploit rich and precious semantic information to derive a well-trained entity linking model under the few-shot setting. The experiments on real-world datasets show that the proposed method can extensively improve the state-of-the-art few-shot entity linking model and achieve impressive performance when only a small amount of labeled data is available. Moreover, we also demonstrate the outstanding ability of the model’s transferability. Our code and models will be open-sourced.

## I. INTRODUCTION

Knowledge bases, such as DBpedia [1], YAGO [2], Freebase [3] and WordNet [4], have been proposed to manage enormous amount of information due to the exponentially increasing of data in recent years [5], [6]. In particular, knowledge bases contain extensive information about the real worlds’ entities, semantic classes, and mutual relationship, which are mostly extracted from the data in the form of natural language. However, the same entity may correspond to different names/words/mentions, and different entities may mean the same name/word/mention in different corpora/domains/sentences, which would cause the ambiguities or errors in knowledge bases. Hence, entity linking, aiming to link

ambiguous mentions to their corresponding entities, has drawn extensive attention from both academia and industry, and massive prominent methods have been proposed [7], [8], [9], [10], [11], [12]. In addition, entity linking can facilitate various tasks such as knowledge base completion [13], [14], [15], question answering [16], [17], [18], and information extraction [19], [20], [21]. For example, many question answering systems, benefiting from the development of entity linking technology, can answer users’ questions more precisely; entity linking can also improve the accuracy of information extraction results due to the consideration of the semantic context. In summary, entity linking is an essential technique for knowledge bases since it resolves the lexical ambiguity of entity mentions and determines their meanings.

The essential challenge of entity linking task is how to embed the mention context and entities more accurately. The early methods based on statistics mainly utilized feature engineering to construct semantic representation [22], [23], [24], [25]. Inspired by the rapid development of deep learning technology, a new generation of approaches for entity linking based on neural models and deep learning has emerged. They have managed to excel in entity linking as in many other natural language processing tasks due to their ability to learn useful deep distributed representations of linguistic data [26], [27], [28], and significant improvements over existing approaches [29], [30], [31] have been observed.

Nevertheless, these methods are still not adequate for real-world entity linking problems. This is because they generally require a large amount of annotation data as the prerequisite. In the absence of sufficient labeled data, the learning ability of these methods will rapidly decline, which significantly restricts the applicability of these methods. Unfortunately, the number of announced data remains far from complete concerning the vast number of real-world entities, as the statistics of each domain shown in Table III. Logeswaran et al. [32] shows that the linking accuracy of a full-transformer model will encounter performance degradation when lacking training data of the corresponding entities in a target domain (see Figure 1). On the other hand, as a main difficulty of entity linking is the

<sup>§</sup> Xiuxing Li and Zhenyu Li contribute equally.

<sup>x</sup> Corresponding author.

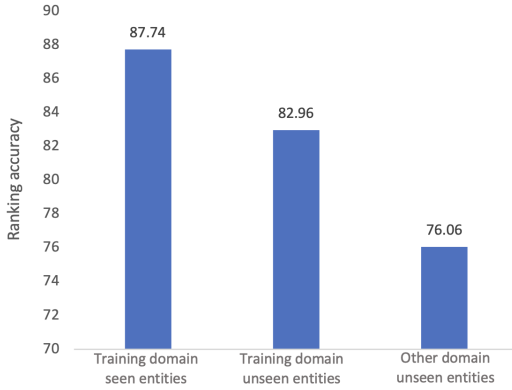


Fig. 1. The performance of existing models degrades dramatically when training data of the corresponding entities and the target domain is insufficient.

lexical ambiguity of entities, most existing algorithms rely on powerful resources of the knowledge base [33], such as alias table [34], frequency statistics [35], entity type [36], etc. However, such powerful resources are not available in specific entity dictionaries (e.g., legal cases, company project descriptions). Most existing methods can't work well and their performance will suffer a severe decline, especially when predicting an entity that has not appeared before.

This makes the study of the few-shot entity linking problem gradually become a new trend [32], [37], [38]. Two different optimization directions have been proposed to alleviate the lack of labeled data, and remarkable results have been achieved: (1) External information enhanced methods take advantage of different sources of external information to improve the quality of data, including manually curated mention tables [39], incoming Wikipedia link popularity [40], and Wikipedia entity categories [37]. Among external information sources, Wikipedia has gradually gained widespread attention from academia. Though external source information can provide supplementary information, these methods inevitably have serious shortcomings. Firstly, introducing external information may introduce new errors; secondly, the algorithm's effectiveness is directly affected by the quality of the external data, and the applicability of the algorithm is inevitably restricted. (2) Pre-trained models, built on large-scale textual data, have involved much knowledge which might be useful for many relevant tasks. Meanwhile, entity linking can obtain a better way to embed the context and entities by leveraging massive general domain knowledge from the pre-trained models like ELMo [41], BERT [42], and their modifications. Recently, Wu et al. presents a two-stage BERT-based linking algorithm BLINK [38] to solve the problem of insufficient data from the technical level by improving the representation ability of the model, which is the most relevant work to us and keeps the state-of-the-art performance in this branch.

However, this direction is still in its infancy and has two main challenges that remained unresolved: (1) They directly use the pre-trained models to represent the context of the

entities. Thus, the quality and quantity of knowledge that they introduced are completely determined by pre-trained models. In other words, the knowledge that they can acquire merely comes from these models, which greatly limits the model's performance. (2) They cannot make adjustments according to the target domain to improve the quality of training data. This is because human language is astoundingly complex and diverse, and there exists a huge gap between different domains.

To deal with the above issues, we present an effective few-shot entity linking system by leveraging a meta-learning mechanism (see Figure 2 for an overview). More specifically, we first adopt a weak supervision strategy to generate in-domain synthetic data. What's more, in the experiment process, we found that it is difficult to control the quality of synthetic data, especially in the specific domains. Therefore, we design a two-stage architecture to generate the synthetic data. Initially, we use a heuristic method to generate the mention-entity pairs. Then we adopt the weak supervision strategy to generate the synthetic data based on these mention-entity pairs. In this way, we can obtain high-quality synthetic data. Furthermore, we design a meta-learning mechanism to improve the quality of the synthetic data. Finally, we conduct extensive experiments on real-world datasets. As shown through detailed experiments, our meta-learning model-based approach achieves much better performance than the state-of-the-art methods on real-world datasets.

The main contributions of this paper are summarized as follows:

- We propose a weak supervision strategy to generate in-domain synthetic data to obtain more knowledge to deal with the ever more severe lack of labeled data.
- Further, we design a meta-learning mechanism to improve the quality of synthetic data by weighting it automatically.
- We propose a novel few-shot named entity linking system to break through the limitation of labeled data.
- We conduct comprehensive experiments on real-world datasets with reasonable measurements, and the results show our model outperforms the state-of-the-art methods.

## II. PRELIMINARY

In this section, we present some preliminary knowledge of the proposed model. Firstly, we list basic notations and definitions in Table I. In the next, we define the relevant underlying elements of the few-shot entity linking problem. Afterward, we define the synthetic data generation process for weak supervision in detail. Finally, considering that meta-learning is an important architectural component of our system, we present a streamlined overview and analysis of the primary approaches to meta-learning.

### A. Entity Linking

Before introducing the entity linking task, we first define some basic concepts.

**Entity.** An entity  $e$  is a meaningful real-world object in a specific knowledge base. For example, an entity in Wikipedia

TABLE I  
NOTATIONS.

Notation	Definition
$G$	A knowledge base.
$E, e$	Entity set of a knowledge base, and $e$ is an entity in knowledge base.
$M, m$	Mention set found in texts, and $m$ is a mention.
$\phi^t$	Model parameters at the $t$ th epoch.
$w$	The weight of samples in the training dataset.
$T5(\tau(e))$	T5 model with summary task.
$L, l_i$	Loss function, specifically, $l_i$ is the loss of $i$ th instance.

can be described by the title and description of the Wikipedia page.

**Mention.** A mention is a textual phrase referring to some entity of a knowledge base such as Unified Medical Language System (UMLS)<sup>1</sup>, YAGO<sup>2</sup> and so on.

In free text, there are a tremendous amount of mentions which can provide useful information. Usually, the mentions in short texts are labeled manually or generated by named-entity recognition methods. Note that mentions may be represented by a short phrase or alias in the short texts.

**Knowledge Base** A knowledge base is a machine-readable source that comprises entities and their relations. We denote a knowledge base as  $G = \{E; R; T\}$ , where  $G$  is a directed graph in which nodes are entities and edges corresponding to the subject-property-object triples,  $E$  indicates a set of entities,  $R$  indicates a set of relations, and  $T$  indicates a set of fact triples. Each triple  $(h, r, t) \in T$  indicates there is a relation  $r \in R$  between  $h \in E$  and  $t \in E$ .

**Domain.** A domain means a specialized entity dictionary such as legal cases and company project descriptions. Each domain has many unique entities related to a specific theme.

**Entity Linking** Given an entity set denoted as  $E$  from the knowledge base  $G$  and a set of texts with mentions  $M$  labeled, the goal of entity linking is to map the mention  $m \in M$  to the  $e \in E$  in the knowledge base  $G$ .

**Few-Shot Entity Linking** The few-shot entity linking is a specific entity link task aimed at accurately linking mentions of free texts to an existing knowledge base with minimal labeled data. Formally, for a knowledge base  $G$  which has  $N$  entities, we only have  $M$  linking examples  $\{(m_i, e_i), 1 \leq i \leq M\}$ , where  $M \ll N$ .

In the above definition of **Entity Linking**, the mentions that emerged in the texts are predefined by various methods,

including entity recognition and other synthetic approaches. The knowledge base here refers to the collection of the entities, their semantic relations, and attributes. Generally, entity linking is the second step after the entity recognition process and can facilitate many tasks. Actually, some mentions may not have corresponding entities in the knowledge base. In our study, we assume that all mentions have their corresponding entities in the knowledge base. Previous studies have shown that entity linking is challenging due to the variations of entity names and entity ambiguity. Many methods have been developed to solve these problems. However, existing entity linking methods require a huge amount of high-quality labeled data, which may be labor-intensive. Unfortunately, high-quality data is generally difficult to access, especially in some specific domains. For example, mentions in the medical domain need to be confirmed by medical experts. In such situation, merely little data can be available. Therefore, there is an urgent need to solve the entity linking problem in the few shot setting. In our paper, the few-shot entity task is aimed at linking the mentions of texts in specific domains without enough labeled data.

### B. Weak Supervision

With sufficient high-quality data, supervised methods can gain comparable success on multiple tasks. In general, the input instance of the supervised learning methods contains two parts: one is the feature vector, and the other is the label. Due to the complexity of real-world learning tasks, high-quality labels are sometimes unavailable. In contrast, noisy and incomplete labels with limited information can be easy to access, which leads to the research of weakly supervised learning [43].

Generally, for supervised learning methods, the best model is trained on high-quality data, which is not realistic when methods are deployed on essential conditions without enough labeled data. Formally, given input data  $X = \{(x_0, y_0), \dots, (x_m, y_m), x_{m+1}, \dots, x_{m+n}\}$ , where  $m$  is the number of labeled data and  $n$  is the number of unlabeled data, the paradigm of weakly supervised learning first generate a pseudo label  $y_i$  for each unlabeled instance  $x_i, m < i \leq m + n$  and reconstruct the training dataset as  $X' = \{(x_0, y_0), \dots, (x_m, y_m), (x_{m+1}, y'_{m+1}), \dots, (x_{m+n}, y'_{m+n})\}$ . Based on the pseudo label, many supervised machine learning algorithms can be applied on the synthetic dataset  $X'$  to learn the predictive models.

### C. Meta-Learning

The meta-learning mechanism is an important architectural component of our few-shot entity linking system. In the following, we will present an uncomplicated overview and analysis of the mainstream approaches of meta-learning. Meta-learning is the scientific approach of systematically observing how different machine learning approaches perform on a wide range of learning tasks and then by learning from this experience, or metadata, to learn new tasks much faster than

<sup>1</sup><https://www.nlm.nih.gov/research/umls/index.html>

<sup>2</sup><https://yago-knowledge.org/>

otherwise possible. Not only does this dramatically speed up and improve the design of machine learning pipelines or neural architectures, but it also allows us to replace hand-engineered algorithms with novel approaches learned in a data-driven way.

Although the continuously evolving deep neural networks have achieved exciting results on various tasks in recent years, existing methods still suffer from diverse deficiencies. For example, humans can learn quickly from a tiny number of samples, while training a neural network with desired performance requires much more data, which leads to its inability to learn quickly. Meta-learning is the method used to solve this problem [44]. In this paper, we mainly take advantage of the meta-learning methodology to adjust the weight of the synthetic data with a small number of samples in the target domain so that the model can quickly adapt to a new domain.

### III. FRAMEWORK

In this section, we introduce the whole framework of our proposed model, which is denoted as MetaBLINK. As shown in Figure 2, MetaBLINK contains two parts: “Target Domain Mention Generation” and “Meta-Learning enhanced Entity Linking”.

#### A. Target Domain Mention Generation.

We leverage the pre-training model T5 [45] to generate many mentions for the target domain in this part. In particular, considering the availability of labeled data in many source domains, we collect the entity-mention pairs (denoted as  $(e, m)$ ) from these source domains and then use them for training the T5 model. In this way, each mention is generated from an entity. To further improve the performance of our model, we also inject knowledge of the target domain into T5 by an unsupervised denoising task. Next, we can leverage the trained T5 model to generate mentions for the given entities in the target domain. Therefore, we can get sufficient synthetic entity-mention pairs for the target domain.

#### B. Meta-Learning enhanced Entity Linking:

Generally, we can learn the entity linking model for the target domain based on the synthetic entity-mention pairs because these data can be regarded as labeled data. However, synthetic data includes both normal cases (i.g., true samples for the target domain) and bad cases (i.g., wrong/meaningless samples for the target domain). Therefore, our goal is to remove bad cases for a better entity linking model. To achieve so, we leverage the meta-learning approach to score each synthetic sample and then iteratively use the weighted samples to retrain the entity linking model. In particular, the meta-learning process needs the few-shot pairs in the target domain to update the weights of synthetic samples. To this end, we can get both better weighted samples and a better trained entity linking model.

### IV. METHODOLOGY

In this section, we give detailed illustration about the MetaBLINK model. As mentioned before, we are aiming at

resolving the following questions: i) how to generate “synthetic samples”; ii) how to use the meta learning to reweight synthetic samples for effectively training an entity linking model.

In particular, the three questions can be answered by the process of “Target Domain Mention Generation”, the process of “Meta-Learning enhanced Entity Linking”, respectively.

#### A. Target Domain Mention Generation

To tackle labeled data lacking problem in the target domain, we first use heuristic methods to get some samples for weak supervision. After that, we adopt a natural language generation model (NLG) to rewrite the mentions to generate more effective samples, which we call “synthetic samples” in the target domain. In this section, we demonstrate the whole procedure in detail.

The prime procedure of synthetic sample generation contains two parts: exact matching and mention rewriting. To reduce the bias introduced in generating labels on the few-shot domain, we first utilize the enormous unlabeled data and employ the exact matching methods to generate the data in the target domain. To further improve the quality of generated data, we adopt the T5 model to generate more semantic-like data samples.

**Exact Matching.** Following Le et al. [46], we generate training samples in the target domain using “Name Matching”. More specifically, if a mention’s name matches an entity’s title, we will link the mention to the corresponding entity in the exact matching step. Through this heuristic method, we can match massive samples in the target domain. Nevertheless, there is a fatal flaw in this way. Suppose we directly train a neural model on these samples. In such case, the powerful models will find a shortcut—straightforward measuring the similarity between mentions and entities’ title to resolve the entity linking problem, rather than measuring the semantic similarity between the context and the entities’ description. These trivial synthetic entity-mention pairs will result in the model being far from adequately trained, which will seriously affect the model’s performance in the prediction process. Though the “exact match” method ensures that mention text matches title text, we find that there are still some errors because an entity may have different mentions.

**Mention Rewriting.** To alleviate the bias mentioned above, we leverage the T5 model. A pre-trained language generation model that can be applied to rewrite the mentions. In this section, we will describe how to train and inference through the T5 model.

Initially, we will give a brief introduction of T5, and more details can be found in [45]. T5 is an encoder-decoder architecture based on the transformer, pre-trained on a large corpus named “C4”. It is a unified model for NLU and NLG tasks. By casting all NLU and NLG tasks to a text-to-text format with different task prefixes, it can give corresponding outputs with the same model.

In our model, we suppose the mention contains part of semantic information of the corresponding entities, so we add

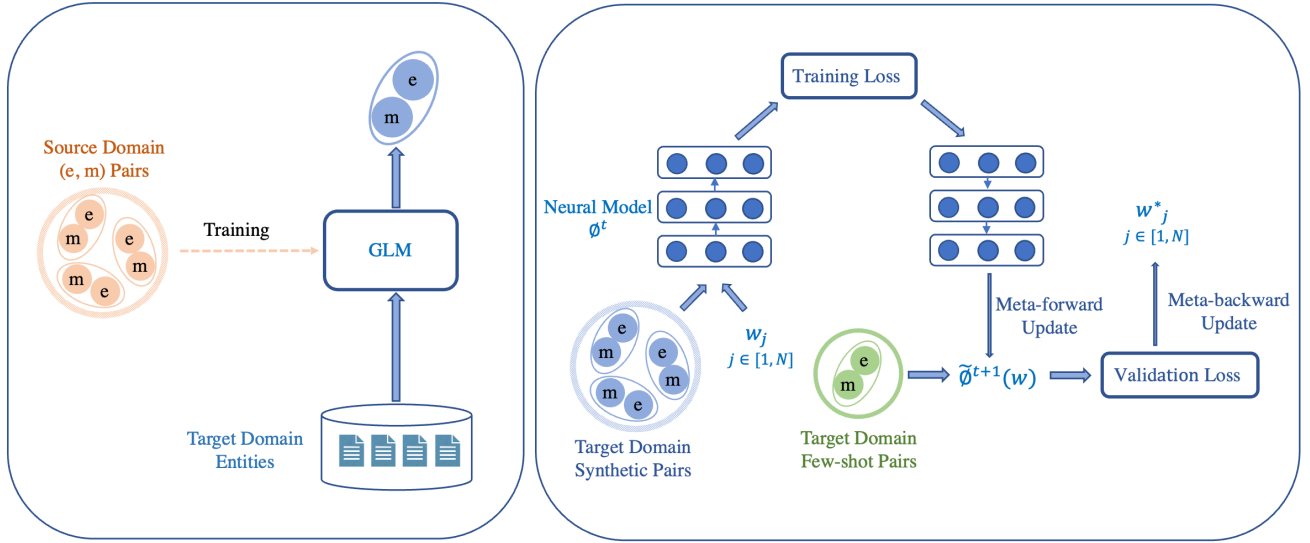


Fig. 2. Illustration of our framework. The left part (a.k.a., Target Domain Mention Generation) focuses on generating large amount of synthetic supervision in target domain. The right part (a.k.a., Meta-Learning enhanced Entity Linking) adopts a meta-learning method to reweight the synthetic data by leveraging the few-shot data in target domain, and learn a neural model based on weighted samples to solve the problem of entity linking.

the prefix “summarize:” to the entity’s description to force the model to summarize the entity in a few words. We regard the mention rewriting procedure as a summarization task. The input is an entity’s description with “summarize:” prefix. The output is a mention. For example, the input is “summarize: The Curse of the Golden Master is the fourth episode of the third season, which was aired on April 16 ...” The output of T5 is “the fourth episode.” Figure 3 is an illustration of the rewriting procedure.

At first, we train T5 in the general domain:

$$m = T5(\tau(e)) \quad (1)$$

where  $e$  is the text description of an entity and  $m$  is its corresponding mention.  $\tau(e)$  is a function that adds the prefix “summarize:” to an entity’s description.

Then we use the trained T5 to rewrite mentions in the target domain:

$$m' = T5(\tau(e)) \quad (2)$$

where  $m'$  is the new mention, which will replace the original mention to form a new context. Note that the original mention has an impact on the generated mention. In the exact match step, mention and entity title are the same. In equation 2, entity title and entity description are used as input.

We designed a fine-tuning strategy that allows T5 to learn semantic information of the target domain. Specifically, we masked the document text with specific sentinel tokens and forced T5 to recover the original text. Through such a procedure, T5 can adapt to a new domain without manual annotation. For example, the sentence “Square Foot is a minifigure released in October 2015” can be masked as “Square  $\langle extra\_id\_0 \rangle$  is a  $\langle extra\_id\_1 \rangle$  released in

$\langle extra\_id\_2 \rangle$ .”

### B. Meta-Learning enhanced Entity Linking

In this section, we first introduce the general process of leverage an deep learning model to solve the entity linking problem, where the model takes as input each synthetic sample (Sec. IV-B1). Next, we leverage the meta learning mechanism to reweight training samples with the help of few real samples in target domain, which is also knows as the denosing process (Sec. IV-B2).

1) *Entity Linking Model*: In the weakly supervised dataset  $\{(m_i, e_i)\}_{i=1}^N$ , each training sample contains two elements, one is the mentions and their contexts, and the other is the entities and their corresponding descriptions. Generally, for mention representation learning, the mention encoder  $ENCODER^m$  takes the mention and its context as input and output the dense representation of the mention. Similarly, for entity embedding, the entity encoder  $ENCODER^e$  takes the entity and the description of the entity as the input and output the dense representation of the entity.

Formally, the encoding process of a mention  $m_i$  and an entity  $e_i$  can be computed in the following way:

$$\mathbf{m}_i = ENCODER^m(m_i, context(m_i)) \quad (3)$$

$$\mathbf{e}_i = ENCODER^e(e_i, desp(e_i)) \quad (4)$$

After that, the match score of a mention  $m_i$  and an entity  $e_i$  can be computed by the cosine distance of their representations:

$$S(m_i, e_i) = \mathbf{m}_i \cdot \mathbf{e}_i \quad (5)$$

To optimise the entity linking model, the loss function is designed to maximize the correct entity-mention pairs with

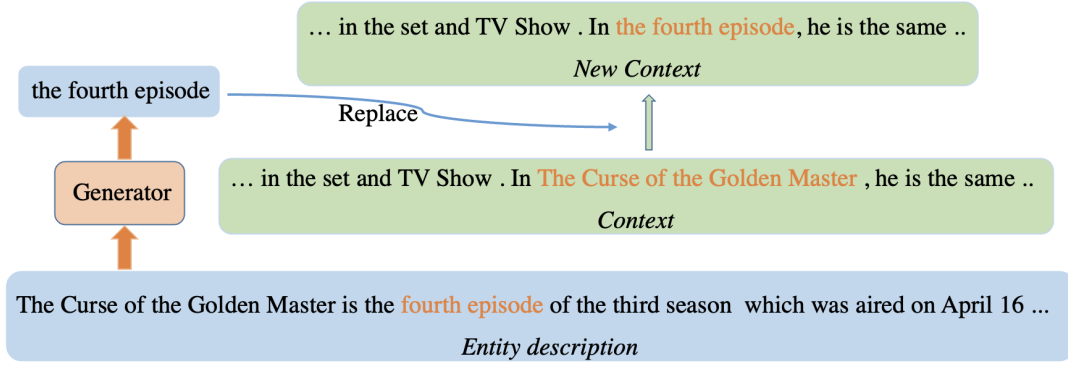


Fig. 3. Example of mention rewriting. In the beginning, the mention “The Curse of the Golden Master” is the same as the entity title. Then we feed the entity’s description into the generator and it generates the new mention “the fourth episode”, which will replace the original mention. The new context and original entity description will form a new training pair.

respect to the (randomly sampled) entities of the same batch  $E(B)[38]$ . Formally, the loss function is defined as follows:

$$l_i(m_i, e_i) = -S(m_i, e_i) + \log \sum_{e \in E(B) \setminus \{e_i\}} \exp(S(m_i, e)) \quad (6)$$

2) *Denoising with Meta-Learning*: Through exact matching and mention rewriting, we have a large number of training instances in the few-shot domain. And in the section IV-B1, we give a deep learning based entity linking framework, which can be used to train on the weakly supervised dataset. However, part of the instances are noisy and may deteriorate the entity linking model’s performance in the target domain. Therefore, we need to find an effective way to select suitable instances for the training process. Previous studies have shown that meta-learning can be used to reweight the instances during the training process automatically and gained relative success in neural ranking [47]. Therefore, we adopt a meta-learning method to reweight these synthetic data in this section.

Formally, to find the best weights for each sample, we need a few target domain samples (seed dataset) to initialize the training process of the weight. Therefore, Given the target domain samples  $\{(m_i, e_i), 1 \leq i \leq M\}$  and the synthetic target domain samples  $\{(m'_j, e'_j), 1 \leq j \leq N\} (M \ll N)$ , the optimization “objective” of entity linking with weighted samples are as the follows:

$$\begin{aligned} \phi(w) &= \arg \min_{\phi} \sum_{j=1}^N w_j^* l_j(m'_j, e'_j; \phi) \\ s.t. \quad w^* &= \arg \min_w \sum_{i=1}^M l_i(m_i, e_i; \phi(w)) \end{aligned} \quad (7)$$

where  $l(m, e; \phi)$  is the loss on  $(m, e)$ .

There are two parameters to be optimized in the training process: the model parameter  $\phi$  and the training instance weights  $w^*$ . To solve the equation 7, following the previous work [47],[48], we divided the optimization stage into three

iterative process: “Meta-forward Update”, “Meta-backward Update” and “Training with New Weight”.

**Meta-forward Update.** Suppose the initial weight for synthetic samples is  $w$ . The meta-forward update optimizes a pseudo neural model parameter  $\hat{\phi}$ :

$$\hat{\phi}(w) = \arg \min_{\phi} \sum_{j=1}^N w_j l_j(m'_j, e'_j; \phi) \quad (8)$$

Note that finding the optimal  $\hat{\phi}$  can be a time-consuming procedure, so we can perform one gradient descent optimization step with a mini-batch sample. Supposing  $\phi^t$  is the parameter at step  $t$  and there exist  $n$  samples in a mini-batch, then  $\hat{\phi}^{t+1}$  can be calculated as:

$$\hat{\phi}^{t+1}(w) = \phi^t - \alpha \frac{\partial}{\partial(\phi^t)} \sum_{j=1}^n w_j l_j(m'_j, e'_j; \phi^t) \quad (9)$$

where  $\alpha$  is the learning rate.

**Meta-backward Update.** In meta-backward update, we can use  $\hat{\phi}(w)$  above to find the optimal weight  $w$ :

$$w^* = \arg \min_w \sum_{i=1}^M l_i(m_i, e_i; \hat{\phi}(w)) \quad (10)$$

Besides, here we can use one gradient descent step to optimize  $w$ :

$$w_j^* = w_j - \alpha \frac{\partial}{\partial(w_j)} \sum_{i=1}^m \frac{1}{m} l_i(m_i, e_i; \hat{\phi}^{t+1}(w)) \quad (11)$$

Following Ren et al. [47], in practice we set the initial weight  $w$  as zero, so the formula can be changed as:

$$\tilde{w}_j = -\eta \frac{\partial}{\partial(w_j)} \sum_{i=1}^m \frac{1}{m} l_i(m_i, e_i; \hat{\phi}^{t+1}(w)) \big|_{w_j=0} \quad (12)$$

Furthermore, to prevent negative weight and make training stable, we drop the negative weight and add normalization

step:

$$\hat{w}_j = \max(0, \tilde{w}_j) \quad (13)$$

$$w_j^* = \frac{\hat{w}_j}{(\sum_{p=1}^n \hat{w}_p) + \delta(\sum_{p=1}^n \hat{w}_p)} \quad (14)$$

where  $\delta(\sum_{p=1}^n \hat{w}_p)$  equals 1 if and only if  $\sum_{p=1}^n \hat{w}_p$  equals 0.

**Training with New Weight.** After getting the new weight  $w^*$ , the neural model can be optimized using standard back propagation:

$$\phi^{t+1} = \arg \min_{\phi^t} \sum_{j=1}^n w_j^* l_j(m'_j, e'_j; \phi^t) \quad (15)$$

In summary, we first calculate the weight of synthetic data based on their impact on few-shot data in the target domain, and then we use the weighted loss to update the neural model.

We depict the training procedure step-by-step in pseudo-code in Algorithm 1, where  $D_f$  denotes the synthetic data and  $D_g$  denotes the few-shot data. And the full training process of MetaBLINK is proposed in Algorithm 2.

## V. IMPLEMENTATION

In this section, we present the implementation of MetaBLINK in detail.

For the reason that MetaBLINK is based on the BLINK[38], most hyper-parameters are set as the same as BLINK: bi-encoder and cross-encoder are optimized with Adam, with the learning rate of 2e-5. The batch size is 128 for the bi-encoder. For cross-encoder, due to meta-learning progress consumes twice as much as initial memory, we can only set the batch size to one. The best model is selected based on the performance of the development set. For the synthetic data generator T5, we select the training domains of the zero-shot entity linking dataset as our training source and train it for 20 epochs with a learning rate of 2e-5. Our model is trained on 2 NVIDIA GeForce RTX 3090 GPUs.

In the few-shot scenario, we directly select 50 samples from the dataset as the seed for meta-learning. We obtain the seed samples in the zero-shot scenario by filtering the synthetic data and using a heuristic self-match method. More details are presented in the next section.

## VI. PERFORMANCE EVALUATION

In this section, we conduct comprehensive experiments on the few-shot entity linking task to evaluate the effectiveness of our model. To start with, we introduce the related datasets and illustrate implementation details. Afterward, we conduct a series of analyses of the experimental results.

### A. Experimental Setup

**Dataset** We conduct extensive experiments on the benchmark used in [32] to evaluate the zero-shot entity linking task. The benchmark dataset is collected from fandom<sup>3</sup>, and the detailed statistics are shown in Table III. Based on the string

<sup>3</sup><https://www.fandom.com/explore-zh>

---

### Algorithm 1 Learning to Reweight Synthetic data

---

**Require:** Synthetic dataset  $D_f$  and seed dataset  $D_g$ , training batch size  $n$ , meta batch size  $m$ , initial model parameter  $\phi_0$

**Ensure:** model parameter at time step  $t$   $\phi_T$

```

1: for  $t = 0 \dots T-1$  do
    // sample a synthetic batch of size  $n$  from synthetic
    dataset
2:    $\{m_f, e_f\} \leftarrow \text{SampleBatch}(D_f, n)$ 
    // sample a seed batch of size  $m$  from the seed set
3:    $\{m_g, e_g\} \leftarrow \text{SampleBatch}(D_g, m)$ 
    // init the weight  $w$  with zero weight and compute the
    loss on the synthetic batch
4:    $w \leftarrow 0; l_f \leftarrow \sum_{j=1}^n w_j L(m_{f,j}, e_{f,j})$ 
    // compute the gradient of  $\phi_t$  with respect to the loss
    of the synthetic batch
5:    $\nabla \phi_t \leftarrow \text{Backward}(l_f, \phi_t)$ 
6:    $\hat{\phi}_t \leftarrow \phi_t - \alpha \nabla \phi_t$ 
    // use the updated model to compute loss on seed batch:
7:    $l_g \leftarrow \frac{1}{m} \sum_{i=1}^m L(m_{g,i}, e_{g,i}; \hat{\phi}_t)$ 
    // calculate the derivative of the loss with respect to the
    initial weight:
8:    $\nabla w \leftarrow \text{Backward}(l_g, w)$ 
    // determine the weight of synthetic batch:
9:    $\tilde{w} \leftarrow \max(-\nabla w, 0); w \leftarrow \frac{\tilde{w}}{\sum_j \tilde{w} + \delta(\sum_j \tilde{w})}$ 
    // truly optimize the model with weighted loss:
10:   $\hat{l}_f \leftarrow \sum_{j=1}^n w_j L(m_{f,j}, e_{f,j}; \phi_t)$ 
11:   $\nabla \phi_t \leftarrow \text{Backward}(\hat{l}_f, \phi_t)$ 
12:   $\phi_{t+1} \leftarrow \text{Optimize}(\phi_t, \nabla \phi_t)$ 
13: end for

```

---



---

### Algorithm 2 MetaBLINK Training Framework

---

**Require:** Source Domain dataset  $D_s$ , seed dataset  $D_g$ , T5, initial MetaBLINK model  $\phi_0$

**Ensure:** best MetaBLINK model  $\phi$

```

1: Generating target domain data  $D_f$ , using Exact Matching
   algorithm using target domain entities.
2: Training T5 with summary task and and augmenting the
   target domain dataset  $D_f$  by rewriting mentions.
3: Training MetaBLINK using Algorithm 1 with  $D_f$ ,  $D_g$  and
    $\phi_0$ .

```

---

overlap between mention and its corresponding entity title, the samples can be divided into four categories: *High Overlap*, *Multiple Categories*, *Ambiguous Substring*, and *Low Overlap*. *High Overlap* refers to the set of samples whose mention text is the same as the title text; *Multiple Categories* is the set of samples whose title text is the mention text followed by a



TABLE II  
EXAMPLES EXHIBITING ERRORS MADE BY THE MODEL TRAINED ON “EXACT MATCH” DATA. THE RED WORDS ARE MENTIONS AND THE ORANGE WORDS IN THE DESCRIPTIONS ARE KEYWORDS THAT ALSO OCCUR IN THE MENTION’S CONTEXT. THE ENTITIES COLORED BY GRAY ARE WRONG ANSWERS.

Mention	Corresponding Entity	Exact Match	Syn
... misty suggested that she partnered with the player and that jack partnered with <b>taku</b> . In the final event, misty asked the player to ...	<b>Yosuke Adachi</b> ... his name is changed to <b>Taku</b> ... Taku and Jack Atlas duel against the <b>Player</b> and <b>Misty</b> ...	Taku	<b>Yosuke Adachi</b>
... sartorius wins easily without even using one turn and he obtains the controls to misgarth’s <b>satellite</b> . jaden wonders what kind of duels are in store for the rest of the ...	<b>SORA (satellite)</b> SORA is the <b>satellite</b> belonging to the country of Misgarth. <b>Sartorius</b> was awarded the keys to this satellite by Prince Ojin upon winning their <b>Duel</b> ...	Satellite (series)	<b>SORA (satellite)</b>
... when <b>his father</b> came to visit, he gave tomoya a zombie figure, telling him that this is zombie and he is the strongest hero in America ...	<b>Mr. Hanasaki</b> Mr. Hanasaki, is Tomoya Hanasaki’s <b>father</b> . Mr. Hanasaki is a “ <b>Zombie</b> ” fan. He works in <b>America</b> ...	The Father We Remember	<b>Mr. Hanasaki</b>

disambiguation phrase, *Ambiguous Substring* refers to the set of samples whose mention text is a substring of the title text, while *Low Overlap* is the set of samples not belonging to the above three categories.

TABLE III  
ZERO-SHOT ENTITY LINKING DATASET.

Dataset	Domain	Entities
#Train	American Football	31,929
	Doctor Who	40,821
	Fallout	16,992
	Final Fantasy	14,044
	Military	104,520
	Pro Wrestling	10,133
	StarWars	87,056
	World of Warcraft	27,677
#Dev	Coronation Street	17,809
	Muppets	21,344
	Ice Hockey	28,684
	Elder Scrolls	21,712
#Test	Forgotten Realms	15,603
	Lego	10,076
	Star Trek	34,430
	YuGiOh	10,031

To evaluate the performance of MetaBLINK under the few-shot circumstances, we further split the samples in each of the four test domains into training/development/test datasets, shown in Table IV. We selected the number of samples among {10, 20, 30, ..., 100}. Specifically, we split off 50 samples as the training set, 50 samples for the development set and keep

the rest as the test set.

TABLE IV  
FEW-SHOT ENTITY LINKING DATASET.

Domain	#Train	#Dev	#Test
Forgotten Realms	50	50	1,100
Lego	50	50	1,099
Star Trek	50	50	4,127
YuGiOh	50	50	3,274

**Evaluation protocol** In the experiment phase, entity linking can be divided into two stages: candidate generation and candidate ranking. We use recall and normalized accuracy (N.Acc.) to measure the performance of the two stages separately. Note that normalized accuracy only considers samples for which the right entity is within the candidates retrieved by the first stage. We use unnormalized accuracy (U.Acc.) to measure the performance of the whole procedure, which is the product of recall and normalized accuracy.

#### B. Few-Shot Entity Linking in Specific Domain

Firstly, we design comprehensive experiments to demonstrate the effectiveness of MetaBLINK when only minimal samples can be available in specific domains. We denote the small amount of data in the target domain as seed. For the synthetic data, we denote the data generated by T5 that is trained only on supervised data on the source domains as syn, and the data generated by T5 that is trained both on supervised data on the source domains and unsupervised data on the target domain as syn\*. Note that the seed was obtained in different ways under different settings. Under the few-shot setting, we



directly split off a few samples as the seed. Since there is no available data on the target domain under the zero-shot domain transfer setting, we use heuristic methods to generate the seed. All syn refers to the same data in the paper. The number of generated data is related to the data scale of a certain domain. Specifically, the numbers of generated data on the Forgotten Realms/Lego/Star Trek/YuGiOh domains are 1656, 1986, 1939, and 1103 respectively. The baselines we compare with MetaBLINK are:

(1) Name Matching, a naive method proposed by Riedel et al. [49] for relation extraction: if a mention matches an entity’s name, then it is regarded as linked to that entity.

(2) BLINK, a strong baseline for zero-shot entity linking task, proposed by Wu et al. [38], which adopts the pre-trained model for both candidates generating stage and candidate ranking stage. To apply BLINK in the few-shot environment, we have made some modifications by training BLINK on the small amount of data in the target domain (seed) and adopt it as our baseline. We also train BLINK on different data sources (syn and syn+seed) for exhaustive comparison.

(3) DL4EL, a denoising method proposed by Le et al. [46], assumes a noise ratio and uses KL divergence to force the model to select high-quality data. We refer readers to Le et al. [46] for detailed description. Note that DL4EL’s denoising procedure is in-batch, while the cross-encoder’s batch size is very small, so we only apply it to the bi-encoder. DL4EL can be applied in almost any architecture, making the model automatically learn to distinguish high-quality data from low-quality data and give larger weights to high-quality data. Morespecificly, we adopted the same bi-encoder and cross-encoder architecture as BLINK in implementing DL4EL.

Results are shown in Table V and Table VI. According to the results, we have the following observations:

(1) Since the majority of samples belong to *Low Overlap* type, the naive name matching method which only relies on shallow surface text doesn’t perform well, indicating that we should integrate context and description information into the model.

(2) Despite BLINK’s powerful reading comprehension ability, merely using the seed set cannot support it to resolve existing problems caused by insufficient labeled data, which reveals the contradiction between the huge-scale model parameters and the minimal number of samples. Comparing it to BLINK trained on the synthetic data, we find that high-quality seed data lead to better cross-encoder accuracy, while synthetic data enhances the bi-encoder significantly. It indicates that cross-encoder is more “quality sensitive,” while a large amount of data can help bi-encoder learn useful patterns, which is important for recall.

(3) DL4EL doesn’t bring improvement as expected. In contrast, it leads to a decline in algorithm performance sometimes. As there is no obvious “bad data” in synthetic data (such as spelling mistakes), DL4EL cannot recognize “bad data” based on simple data features. At the same time, due to the lacking of high-quality seed data, the model cannot distinguish useful data well. In contrast, MetaBLINK can combine the strength of

synthetic data and seed data. Moreover, using the seed data as extra supervision, the model can make full use of the synthetic data and achieve the best performance, which gains significant improvements compared to name matching or directly fine-tuned models.

(4) Experiments showed that this procedure could slightly improve T5’s rewriting ability. MetaBLINK gains further improvement by leveraging using syn\* data compared to syn data. The experimental results indicate that the additional unsupervised denoising training step can improve the rewriting ability of T5 via generating more fluent mentions with fewer errors.

### C. Zero-Shot Domain Transfer

In this section we evaluate our method on a more challenging setting: zero-shot entity linking. Logeswaran et al. [32] present the zero-shot entity linking task, where they train the model on 8 training domains and then directly evaluate the model on 4 testing domains without the aid of any in-domain samples. We didn’t put syn + seed because the 8 training domains already have high quality and plenty of data. Note that introducing the term general domain is to distinguish train set from specialized target domains. In addition, we found that BLINK pretrained on Wikipedia underperformed BLINK trained only on the train set.

First, we use heuristic methods to generate the seed set instead of directly splitting it from the original dataset. Specifically, we generate high-quality seed data in two strategies:

(1) Filtering the synthetic data by some rules, such as correct spelling, no overlapping between mention and entity title, etc.

(2) Self-match: *Multiple Categories* type occupies a large proportion in the dataset, while it is rare in our synthetic data. To fill this vacancy, we focus on entities whose title contains a disambiguation phrase. We find the mention in the entity’s own description which is the same as the title (excluding the disambiguation phrase). The results are reported in Table VII.

The experimental results show that our model can achieve better performance on testing domains compared to BLINK. Moreover, the results are diverse across domains. In the Forgotten Realms domain and Star Trek domain, MetaBLINK only achieves little progress, while in the Lego domain and Yugioh domain, it leads to obvious improvement. We suppose it is because the “gaps” between the general domain and testing domains are different. For example, the gap between the Star Trek domain and the general domain is small. Thus, it is tough to get improvement on Star Trek by leveraging pre-trained models.

To demonstrate our hypothesis, we design a tiny experiment to test the “gap”. Specifically, for every testing domain, we select 500 samples for training, allowing models to be exposed to domain knowledge sufficiently. Then we regard the improvement of accuracy as the “gap” between this domain and general domain. Results are shown in Table VIII. BLINK+FT in the table means BLINK trained on the 500 samples. The results show that the gap between the general domain and the Forgotten Realms/Star Trek domain is relatively small,

TABLE V  
U.ACC. ON FORGOTTEN REALMS DOMAIN AND LEGO DOMAIN. SYN REPRESENTS THE SYNTHETIC DATA AND SEED REPRESENTS THE FEW-SHOT DATA IN TARGET DOMAIN.

Method	Data	Forgotten Realms			Lego		
		R@64	N.Acc.	U.Acc.	R@64	N.Acc.	U.Acc.
Name Matching [49]	-	-	-	19.64	-	-	12.37
BLINK [38]	Seed	35.27	<b>59.02</b>	20.82	52.68	45.60	24.02
BLINK	Syn	63.82	40.33	25.74	72.88	28.59	20.83
BLINK	Syn+Seed	66.75	54.10	36.11	75.52	48.80	36.85
DL4EL [46]	Syn+Seed	66.71	54.10	36.09	75.12	48.80	36.65
MetaBLINK	Syn+Seed	68.91	56.33	38.82	77.80	50.18	39.04
MetaBLINK	Syn*+Seed	<b>69.56</b>	56.27	<b>39.14</b> (↑ 18.32)	<b>78.17</b>	<b>50.65</b>	<b>39.59</b> (↑ 15.57)

TABLE VI  
U.ACC. ON STAR TREK DOMAIN AND YUGIOH DOMAIN.

Method	Data	Star Trek			Yugioh		
		R@64	N.Acc.	U.Acc.	R@64	N.Acc.	U.Acc.
Name Matching [49]	-	-	-	12.12	-	-	7.88
BLINK [38]	Seed	21.57	<b>37.08</b>	8.00	35.00	37.70	13.20
BLINK	Syn	55.61	21.31	11.85	55.77	22.84	12.74
BLINK	Syn+Seed	59.82	32.14	19.23	60.72	35.11	21.32
DL4EL [46]	Syn+Seed	59.91	32.14	19.26	59.20	35.11	20.79
MetaBLINK	Syn+Seed	61.09	34.51	21.08	60.72	37.58	22.82
MetaBLINK	Syn*+Seed	<b>61.41</b>	34.63	<b>21.27</b> (↑ 9.15)	<b>61.28</b>	<b>38.02</b>	<b>23.30</b> (↑ 10.10)

TABLE VII  
U.ACC. ON FOUR DOMAINS WHEN DOING ZERO-SHOT DOMAIN TRANSFER.

Method	Data	Forgotten Realms	Star Trek	Lego	Yugioh
BLINK	-	84.11	74.45	72.22	66.30
BLINK	Seed	84.60	74.51	73.51	68.80
MetaBLINK	Syn+Seed	<b>84.81</b>	<b>74.54</b>	<b>74.11</b>	<b>69.50</b>

TABLE VIII  
GAPS BETWEEN GENERAL DOMAIN AND FOUR TESTING DOMAINS MEASURED BY U.ACC. DIFFERENCE.

Method	Forgotten Realms	Star Trek	Lego	Yugioh
BLINK	85.19	74.49	71.79	65.61
BLINK+FT	88.55	77.04	78.46	73.08
GAP	3.36	2.55	6.67	7.47

indicating that the model trained on the general domain already acquires the abilities required in these domains. On the other hand, the gap in Lego and Yugioh domains is relatively large, indicating that the model has much potential. Correspondingly, our MetaBLINK can achieve 2-4% improvement in the Lego domain and Yugioh domain, demonstrating that our method can help models bridge the gap between different domains.

We summarize our design thought behind the experiment

for zero domain transfer as follows: Table VII shows the results of MetaBlink in four target domains. The improvement on Forgotten Realms and Star Trek domains is not obvious. We suspect this may be because the gaps between these two domains and the general domain are relatively small. On this basis, we experimented with Table VIII to calculate the gap between target and general domains, and the experimental results confirmed our conjecture. Since Table IX is to verify MetaBLINK’s domain transferability, we abandoned Forgotten Realms and Star Trek domains for the zero-shot domain transfer experiment.

In the aforementioned experimental phase, we utilize synthetic data as our training source, while in a zero-shot setting, we can take advantage of general domain data. Table IX shows the domain transfer results using different training sources.

From the results, we can find that the general domain data and synthetic data can both improve the transferability

TABLE IX  
U.ACC. ON TWO DOMAINS WHEN DOING ZERO-SHOT DOMAIN TRANSFER  
USING DIFFERENT TRAINING SOURCES.

Method	Data	Lego	Yugioh	Avg
BLINK	-	72.22	66.30	69.26
BLINK	Seed	73.51	68.80	71.16
MetaBLINK	Syn+Seed	74.11	69.50	71.81
MetaBLINK	General+Seed	74.82	68.90	71.89
MetaBLINK	General+Syn+Seed	<b>74.90</b>	69.52	72.21
MetaBLINK	General+Syn*+Seed	<b>74.90</b>	<b>69.55</b>	<b>72.23</b>

of model. We suppose the general domain data has higher quality while synthetic data introduces more domain-specific knowledge, and the distinction between syn and syn\* data can be neglected when having plenty of general domain data. Jointly using different sources can lead to nearly 3 points average improvement.

#### D. The Effect of Meta Learning

In this section, we confirm the necessity of using meta-learning. We generate bad samples by linking mentions to random entities and test whether MetaBLINK can distinguish bad data from normal data. Figure 4 shows the result, with the y-axis representing the selected ratio. We can find that by the supervision of the seed set, the model select nearly 50% of the normal data, but only 20% of the bad data, which indicating meta-learning plays an important role in the whole procedure.

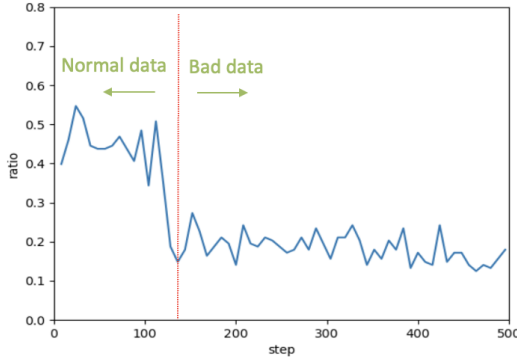


Fig. 4. Effectiveness of meta learning: we firstly generate bad training data by linking mentions to random entities, then we record selecting ratio of different data source when supervised by few-shot data in target domain. The experiment is done on bi-encoder, Yugioh domain.

#### E. The Effect of Mention Rewriting

In this part, we further design a series of additional experiments to investigate the effectiveness of mention rewriting on generating synthetic data. Concretely, we use Exact Match to represent the synthetic data that is generated only based on the strategy of exact matching. Firstly, we train BLINK on the different data sources and compare their corresponding linking accuracy. The results are shown in Table X. As expected, the syn data can effectively alleviate the bias and

improve performance both on recall and ranking accuracy, and syn\* achieves slight improvement in most cases by using an additional text denoising procedure. Table II shows some typical errors made by the model trained on “Exact Match” data.

Besides linking performance, we want to statistically analyze the difference between T5 generated data and Exact Match data. Specifically, we experiment to demonstrate that T5 generated data is closer to natural data distribution than Exact Match data. We introduce the ROUGE<sup>4</sup> metric to compare the golden mentions’ similarity with T5 generated mentions and exact match mentions respectively. The golden mentions are sampled from the dataset of a specific domain. ROUGE metric is often used in the field of machine translation to evaluate the quality of abstract generation. As the length of generated mentions is relatively short, we adopted ROUGE-1 as the primary metric here. Table XI shows the F1 score between golden mentions and different data sources. The results prove that the mentions generated by T5 are closer to the data distribution of the target domain.

## VII. RELATED WORK

We address the problem of few-shot entity linking. Prior attempts to make up insufficient annotation data are all subject to one or more of the following limitations:

- The process of introducing external knowledge will inevitably incur additional costs, and introducing external information may even introduce new errors.
- They depend excessively on external data, the effectiveness of the algorithms is directly affected by the quality of the external data, and the their utility is indispensable restricted.
- Neither can they generate the knowledge of the target domain nor improve the data quality by making adjustments according to the target domain.

Based on the techniques leveraged, existing few-shot entity linking optimization strategies can be divided into two categories: external information enhanced strategy and pre-trained model-based strategy. In this section, we briefly summarize the related works from these two aspects: external information enhanced entity linking and pre-trained model-based entity linking.

#### A. External information enhanced entity linking

External information enhanced methods take advantage of different sources of external information to improve the quality of data, including manually curated mention tables [39], incoming Wikipedia link popularity [40], and Wikipedia entity categories [37]. Among external information sources, Wikipedia has gradually gained widespread attention from academia. Though external source information can provide supplementary semantic information, these methods inevitably have serious shortcomings. Firstly, introducing external information may introduce new errors; secondly, the performance

<sup>4</sup><https://github.com/huggingface/datasets/blob/master/metrics/rouge/rouge.py>

TABLE X  
EFFECTIVENESS OF MENTION REWRITING.

Domain	Lego		Yugioh		Forgotten Realms		Star Trek	
Metric	R@64	N.Acc.	R@64	N.Acc.	R@64	N.Acc.	R@64	N.Acc.
Exact Match	72.07	25.76	49.54	20.56	60.08	38.46	54.22	20.74
Syn	72.88	28.59	55.77	22.84	63.82	<b>40.33</b>	55.61	21.31
Syn*	<b>73.21</b>	<b>29.03</b>	<b>56.32</b>	<b>23.36</b>	<b>64.61</b>	40.20	<b>55.71</b>	<b>21.36</b>

TABLE XI  
F1 SCORE OF ROUGE-1 USING DIFFERENT METHODS.

Domain	Lego	Yugioh	Forgotten Realms	Star Trek
Exact Match	33.70	38.01	40.18	28.85
Syn	42.91	45.90	42.26	33.98
Syn*	<b>43.96</b>	<b>46.56</b>	<b>42.98</b>	<b>34.03</b>

of the algorithm is directly affected by the quality of the external data, and the applicability of the algorithm is indispensable restricted.

There are recent works demonstrating that fine-grained entity typing information enhances the linking progress [50], [51], [52]. [32] proposed the few-shot entity linking task. They use cross encoders for entity ranking, but rely on traditional IR-techniques for candidate generation and do not evaluate on large-scale benchmarks such as TACKBP [37] show that dense embeddings work well for candidate generation. However, they do not involve pre-training or improving the quality of the generated data but include external category labels in their bi-encoder architectures, which limits their linking to entities in Wikipedia. Our approach can be seen as combining the backbones of existing works and demonstrate for the first time that pre-trained few-shot architectures are both highly accurate and computationally efficient at scale.

### B. Pre-trained model-based entity linking

The study of pre-trained model-based few-shot entity linking is still in its infancy. While some pioneering works have been proposed, they used the BERT-based model directly to represent entities and related information. Wu et al. present a two-stage BERT-based linking algorithm [38]. The first stage does retrieval in a dense space defined by a bi-encoder that independently embeds the context and the entity descriptions. Then each candidate is re-ranked with a cross-encoder that concatenates the mention and entity text. However, neither can it generate knowledge of the target domain nor make adjustments according to the target domain to resolve the ambiguity of entities. It is the most relevant work to us and keeps the state-of-the-art performance in this branch.

## VIII. FUTURE WORK

In future work, we will work on investigating the remaining bottlenecks in MetaBLINK’s performance. We next plan to focus on the process of noise data generation; in addition,

MetaBLINK might benefit from meta-learning mechanism improvements and supplementary information. In MetaBLINK, the embedding progress of entity context leaves little room for further improvement. We will subdivide the entity mentions and make statistics on the accuracy of different categories to conduct a more deeply exploration in our further work.

We also expect that few-shot entity linking will rapidly develop and attract more enriching features like global coherence across all entities in a document, NIL prediction, joining entity retrieve and entity linking steps together, or providing complete end-to-end solutions. The latter would be an incredibly challenging task but also a fascinating research direction. Furthermore, performing named entity recognition and entity linking jointly to make these two tasks reinforce each other is a promising direction, especially for text where the named entity recognition tools perform poorly. Finally, we will explore to improve the quality of generated mention via leveraging the gap between the source and target domain.

Our long-term vision is to design a domain-specific entity linking model to discover more entities and support various downstream applications.

## IX. CONCLUSION

In this paper, we conduct a comprehensive study of few-shot entity linking techniques, which is inevitable and challenging in the real world. To obtain more knowledge to deal with the ever more severe lack of labeled data, we propose a weak supervision strategy to generate in-domain synthetic data. To further differentiate the quality of each synthetic data instance for model training, we design a meta-learning mechanism that can automatically assign different weights to the synthetic data. We conduct comprehensive experiments on real-world datasets with reasonable measurements, and the results show our approach can significantly improve the state-of-the-art methods.

## ACKNOWLEDGMENT

This work was supported in part by National Key Research and Development Program of China under Grant No. 2020YFA0804503, National Natural Science Foundation of China under Grant No. 61521002, Beijing Academy of Artificial Intelligence (BAAI), and Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education. We further thank all anonymous reviewers for their constructive comments, which help us to improve the quality of this manuscript.

## REFERENCES

- [1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web*, K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735.
- [2] M. Fabian, K. Gjergji, W. Gerhard *et al.*, "Yago: A core of semantic knowledge unifying wordnet and wikipedia," in *16th International World Wide Web Conference, WWW*, 2007, pp. 697–706.
- [3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.
- [4] G. A. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.
- [5] X. Qin, Y. Luo, N. Tang, and G. Li, "Making data visualization more efficient and effective: a survey," *VLDB J.*, vol. 29, no. 1, pp. 93–117, 2020.
- [6] Y. Luo, X. Qin, N. Tang, and G. Li, "Deepeye: Towards automatic data visualization," in *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*. IEEE Computer Society, 2018, pp. 101–112. [Online]. Available: <https://doi.org/10.1109/ICDE.2018.00019>
- [7] G. Mann and D. Yarowsky, "Unsupervised personal name disambiguation," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 2003, pp. 33–40.
- [8] J. Artilles, J. Gonzalo, and S. Sekine, "The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task," in *Proceedings of the fourth international workshop on semantic evaluations (semeval-2007)*, 2007, pp. 64–69.
- [9] X. Yin, J. Han, and S. Y. Philip, "Object distinction: Distinguishing objects with identical names," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 1242–1246.
- [10] X. Han and J. Zhao, "Named entity disambiguation by leveraging wikipedia semantic knowledge," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 215–224.
- [11] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran, "Evaluating entity linking with wikipedia," *Artificial intelligence*, vol. 194, pp. 130–150, 2013.
- [12] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks," in *SIGMOD '21: International Conference on Management of Data, China, June 20-25, 2021*, G. Li, Z. Li, S. Idreos, and D. Srivastava, Eds. ACM, 2021, pp. 1235–1247. [Online]. Available: <https://doi.org/10.1145/3448016.3457261>
- [13] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 601–610.
- [14] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer *et al.*, "Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015.
- [15] B. Shi and T. Wenginger, "Open-world knowledge graph completion," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [16] D. Diefenbach, V. Lopez, K. Singh, and P. Maret, "Core techniques of question answering systems over knowledge bases: a survey," *Knowledge and Information systems*, vol. 55, no. 3, pp. 529–569, 2018.
- [17] W. Yin, M. Yu, B. Xiang, B. Zhou, and H. Schütze, "Simple question answering by attentive convolutional neural network," *arXiv preprint arXiv:1606.03391*, 2016.
- [18] A. P. B. Veyseh, "Cross-lingual question answering using common semantic space," in *Proceedings of TextGraphs-10: the workshop on graph-based methods for natural language processing*, 2016, pp. 15–19.
- [19] F. Hasibi, K. Balog, and S. E. Bratsberg, "Exploiting entity linking in queries for entity retrieval," in *Proceedings of the 2016 acm international conference on the theory of information retrieval*, 2016, pp. 209–218.
- [20] F. Corcoglioniti, M. Dragoni, M. Rospocher, and A. P. Aprosio, "Knowledge extraction for information retrieval," in *European Semantic Web Conference*. Springer, 2016, pp. 317–333.
- [21] F. Hasibi, K. Balog, and S. E. Bratsberg, "Entity linking in queries: Tasks and evaluation," in *Proceedings of the 2015 international conference on the theory of information retrieval*, 2015, pp. 171–180.
- [22] Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, and H. Wang, "Learning entity representation for entity disambiguation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2013, pp. 30–34.
- [23] D. Newman-Griffis, A. M. Lai, and E. Fosler-Lussier, "Jointly embedding entities and text with distant supervision," *arXiv preprint arXiv:1807.03399*, 2018.
- [24] P. Radhakrishnan, P. Talukdar, and V. Varma, "Elden: Improved entity linking using densified knowledge graphs," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1844–1853.
- [25] Y. Luo, X. Qin, C. Chai, N. Tang, G. Li, and W. Li, "Steerable self-driving data visualization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 475–490, 2022.
- [26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.
- [27] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [28] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Natural language to visualization by neural machine translation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 217–226, 2022.
- [29] A. Chisholm and B. Hachey, "Entity disambiguation with web links," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 145–156, 2015. [Online]. Available: <https://aclanthology.org/Q15-1011>
- [30] Z. Kraljevic, T. Searle, A. Shek, L. Roguski, K. Noor, D. Bean, A. Mascio, L. Zhu, A. A. Folarin, A. Roberts *et al.*, "Multi-domain clinical natural language processing with medcat: the medical concept annotation toolkit," *Artificial Intelligence in Medicine*, vol. 117, p. 102083, 2021.
- [31] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 1375–1384.
- [32] L. Logeswaran, M. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Marquez, Eds. Association for Computational Linguistics, 2019, pp. 3449–3460. [Online]. Available: <https://doi.org/10.18653/v1/p19-1335>
- [33] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 509–518.
- [34] T. Févry, N. FitzGerald, L. B. Soares, and T. Kwiatkowski, "Empirical evaluation of pretraining strategies for supervised entity linking," *arXiv preprint arXiv:2005.14253*, 2020.
- [35] V. I. Spitkovsky and A. X. Chang, "A cross-lingual dictionary for english wikipedia concepts," 2012.
- [36] S. Cucerzan, "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)*, 2007, pp. 708–716.
- [37] D. Gillick, S. Kulkarni, L. Lansing, A. Presta, J. Baldridge, E. Ie, and D. Garcia-Olano, "Learning dense representations for entity retrieval," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 528–537. [Online]. Available: <https://aclanthology.org/K19-1049>
- [38] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, "Scalable zero-shot entity linking with dense entity retrieval," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6397–6407. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.519>

- [39] O.-E. Ganea and T. Hofmann, “Deep joint entity disambiguation with local neural attention,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2619–2629. [Online]. Available: <https://aclanthology.org/D17-1277>
- [40] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy, “Named entity disambiguation for noisy text,” in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 58–68. [Online]. Available: <https://aclanthology.org/K17-1008>
- [41] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: <https://aclanthology.org/N18-1202>
- [42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://aclanthology.org/N19-1423>
- [43] Z.-H. Zhou, “A brief introduction to weakly supervised learning,” *National science review*, vol. 5, no. 1, pp. 44–53, 2018.
- [44] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, “Meta-learning in neural networks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [45] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020. [Online]. Available: <http://jmlr.org/papers/v21/20-074.html>
- [46] P. Le and I. Titov, “Distant learning for entity linking with automatic noise detection,” *arXiv preprint arXiv:1905.07189*, 2019.
- [47] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 4331–4340. [Online]. Available: <http://proceedings.mlr.press/v80/ren18a.html>
- [48] S. Sun, Y. Qian, Z. Liu, C. Xiong, K. Zhang, J. Bao, Z. Liu, and P. Bennett, “Meta adaptive neural ranking with contrastive synthetic supervision,” *arXiv preprint arXiv:2012.14862*, 2020.
- [49] S. Riedel, L. Yao, and A. McCallum, “Modeling relations and their mentions without labeled text,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.
- [50] J. Raiman and O. Raiman, “Deeptype: multilingual entity linking by neural type system evolution,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [51] Y. Onoe and G. Durrett, “Fine-grained entity typing for domain independent entity linking,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8576–8583.
- [52] S. Khalife and M. Vazirgiannis, “Scalable graph-based method for individual named entity identification,” in *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 17–25. [Online]. Available: <https://aclanthology.org/D19-5303>