A Transformational Biencoder with In-Domain Negative Sampling for Zero-Shot Entity Linking

Kai Sun,[1] **Richong Zhang,**[1*] **Samuel Mensah,**[2] **Yongyi Mao,**[3] **Xudong Liu**[1]
[1]SKLSDE, Beihang University, Beijing, China
[2]Department of Computer Science, University of Sheffield, UK
[3] School of Electrical Engineering and Computer Science, University of Ottawa, Canada
sunkai@buaa.edu.cn, s.mensah@sheffield.ac.uk, ymao@uottawa.ca
zhangrc,liuxd@act.buaa.edu.cn

Abstract

Recent interest in entity linking has focused in the zero-shot scenario, where at test time the entity mention to be labelled is never seen during training, or may belong to a different domain from the source domain. Current work leverage pre-trained BERT has the implicit assumption that BERT bridges the gap between the source and target domain distributions. However, fine-tuned BERT has a considerable underperformance at zero-shot when applied in a different domain. We solve this problem by proposing a Transformational Biencoder that incorporates a transformation into BERT to perform a zero-shot transfer from the source domain during training. As like previous work, we rely on negative entities to encourage our model to discriminate the golden entities during training. To generate these negative entities, we propose a simple but effective strategy that takes the domain of the golden entity into perspective. Our experimental results on the benchmark dataset Zeshel show effectiveness of our approach and achieve new state-of-the-art.

1 Introduction

Entity Linking (EL) is an important task in Natural Language Processing (NLP), which seeks to align entity mentions in a document to their referent entity in a knowledge base such as Wikipedia. EL has received widespread attention due to its application in a variety of tasks, including information extraction (Lin et al., 2012), knowledge base population (Dredze et al., 2010), content analysis (Weng et al., 2010), etc. There has been great achievement in building EL systems, however, majority of proposed works (Ganea and Hofmann, 2017; Cao et al., 2018) are built on the assumption that the entity set is shared among the train and test sets. In many practical cases, however, the train and test sets may come from different domain distributions. This potentially creates disjoint entity sets across the different domains, highlighting the importance of zero-shot EL (Sil et al., 2012; Logeswaran et al., 2019).

Zero-shot EL aims to label mentions in the test set that have never been seen during training. A line of works have proposed zero-shot learning techniques for entity linking (Sil et al., 2012; Wang et al., 2015). The common paradigm in these works is to link labelled mentions in a document to entities in well structured knowledge bases. Despite their promising successes, labelled data is typically expensive to produce or are not easily obtained for some specialized domains such as the legal domain. To enable research in this problem, Logeswaran et al. (2019) developed the Zeshel dataset which contains a diverse range of specialized domains, in which mentions and entities have rich textual content. Without adopting resources (e.g., structured knowledge base) or assumptions (e.g., labelled mentions, a shared entity set), they expand the scope of zero-shot EL to promote the generalizability of EL system on unseen domains.

So far, only few works have been proposed (Wu et al., 2020; Yao et al., 2020; Zhang and Stratos, 2021; Tang et al., 2021), where BERT (Devlin et al., 2018) is found to be the notable encoder. Majority of these works are devoted to retrieving candidate entities since this is essential to candidate entity ranking for EL systems. Zhang and Stratos (2021) achieved state-of-the-art for candidate entity generation by employing the Biencoder (Wu et al., 2020) to encode mentions and entities, an expressive Sum-Of-Max (SOM) score function (Khattab and Zaharia, 2020) to compute their relevance scores, and by optimizing with hard negative entities. To generate hard negatives, Zhang and Stratos (2021) use the score function to rank all entities across domains and select top-k entities.

Although the Biencoder has been successfully applied, it faces a fundamental weakness which limits its ability to successfully achieve zero-shot

[*] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2022, pages 1449 - 1458
May 22-27, 2022

transfer for the task. Specifically, the fine-tuned BERT as used in the Biencoder has been shown to degrade substantially on zero-shot transfer if there is a shift between source and target domains (Ma et al., 2019). Another problem is the high dimensionality of entities/mentions which poses a challenge for complex scorers (e.g., Sum-Of-Max). SOM requires $\mathcal{O}(n^2)$ in running time and storage complexity, which makes it almost infeasible when sampling hard negatives. As EL systems may have millions of entities, a more scalable solution is needed.

In this paper, we overcome the aforementioned weakness of the Biencoder by integrating it with a learnable transformation, developing an Transformational Bi-encoder (T-Biencoder). As the name suggests, we focus on learning a transformation in the BERT architecture to achieve zero-shot transfer from a source domain during training. With regard to the efficiency of the model and the optimization strategy, we recognize the performance advantages of sampling with hard negatives over sampling randomly. Hard negative sampling works because generated entities are semantically different and close to the golden entity in the embedding space. This encourages discrimination between the golden entity and negative entities. We hypothesize that, the condition of lexical similarity between the entity and golden entity will lead to harder negatives and better optimization. That is, we propose to random (or hard) sample in-domain of the golden entity. By sampling in-domain, the entity set in which we sample from is relatively small, making it a more efficient alternative to sampling across domains. Extensive experiments show the effectiveness of our approach as against prior works.

Our contributions can be summarized as follows:

- we propose a Transformational Biencoder, which incoporates a transformation into the Biencoder (Wu et al., 2020) to improve generalization on unseen domains for zero-shot EL.

- we propose in-domain negative sampling to encourage our model to discriminate the golden entity, which in effect improves optimization and efficiency.

- we perform extensive experiments to demonstrate the effectiveness of our approach, and achieve state-of-the-art on the Zeshel dataset (Logeswaran et al., 2019).

2 Related Works

To enable progress on the zero-shot entity linking task, Logeswaran et al. (2019) propose to use the naive baseline method BM25 (Robertson and Zaragoza, 2009) to measure the relevance score of mention-entity pairs. Following this work, a number of methods operating on Zeshel have been proposed (Wu et al., 2020; Yao et al., 2020; Zhang and Stratos, 2021; Tang et al., 2021), where BERT (Devlin et al., 2018) is found to be the notable encoder. This is not surprising as BERT has shown to produce state-of-the-art results in several NLP tasks. Among the existing works, Wu et al. (2020) propose a Biencoder architecture where two independent BERT encoders are employed to encode the textual descriptions of mentions and entities. A dot product is used as the scorer, referred to as DUAL by (Zhang and Stratos, 2021). The Biencoder provides a strong baseline for the task due to the expressiveness of BERT. Yao et al. (2020) adapts a BERT architecture that repeats the position embedding to solve the long-range modeling problem in entity textual descriptions. Tang et al. (2021) propose a bidirectional multi-paragraph reading model that leverages more textual information to enhance text understanding capability. Zhang and Stratos (2021) adopt the Biencoder framework but employ a more expressive Sum-Of-Max score function (Khattab and Zaharia, 2020) to measure the relevance between a mention and entity, achieving state-of-the-art results on the task. The majority of these works leverage negative entities during optimization. They also have an implicit assumption that BERT is sufficient for zero-shot transfer.

Unlike previous work, we adapt the BERT encoder with a transformation to improve zero-shot transfer. We also consider to sample negative entities in-domain of the golden entity rather than across all domains to improve optimization and efficiency.

3 Methodology

In this section, we describe Transformational Biencoder (T-Biencoder), our proposed method for zero-shot EL. First, we formally present the task definition in Section 3.1. Next, in Section 3.2 we introduce the Biencoder (Wu et al., 2020). Then, we describe our adaptation of the Biencoder to develop T-Biencoder in Section 3.3. Finally, we close by discussing our negative sampling strategy in Section 3.4.

3.1 Task Definition

The entity linking task is formulated as follows. Given a mention m in a document and a set of entities $\gamma = \{e_i\}_{i=1,\dots,N}$, EL aims to identify the referent entity $e \in \gamma$ corresponding to mention m. The goal is to obtain an EL model on a train set of mention-entity pairs $D^{\text{Train}} = \{(m_i, e_i)|e_i \in \gamma\}_{i=1,\dots,M}$, that correctly labels mentions in the test set D^{Test}. D^{Train} and D^{Test} are typically assumed to come from the same domain.

In this paper, we focus on the zero-shot EL (Logeswaran et al., 2019), where both $D^{\text{Train}} = \{D^i_{\text{src}}\}_{i=1,\dots,N_{\text{src}}}$ and $D^{\text{Test}} = \{D^i_{\text{tgt}}\}_{i=1,\dots,N_{\text{tgt}}}$ are found to contain multiple sub-datasets from different domains. Note that the entity sets $\gamma^1_{\text{src}}, \dots, \gamma^{N_{\text{src}}}_{\text{src}}, \gamma^1_{\text{tgt}}, \dots, \gamma^{N_{\text{tgt}}}_{\text{tgt}}$ corresponding to the sub-datasets are disjoint, with entities or mentions expressed as textual descriptions. Our goal is to build a model upon D^{Train} to label mentions in D^{Test}.

3.2 Biencoder

Our model is built on the Biencoder (Wu et al., 2020), that independently embeds the mention and corresponding entity in the same representation space. As shown in Figure 1, the Biencoder consists of a text encoder E_{θ_m} for encoding mentions, a text encoder E_{θ_e} for encoding entities and a score function f to compute a relevance score of mention-entity pairs. E_{θ_m} and E_{θ_e} share the same architecture but have independent parameters, θ_m and θ_e. BERT (Devlin et al., 2018) is employed to model E_{θ_m} and E_{θ_e}.

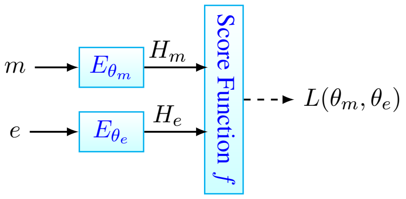


Figure 1: Architecture of Biencoder

Given the mention-entity pair (m, e), the mention m is characterized by the left context (ctx$_l$) and right context (ctx$_r$) of the mention as well as the mention itself. Thus, m is represented as the BERT input:

$$m = \text{[CLS] ctx}_l \text{ [Ms] mention [Me] ctx}_r \text{ [SEP]} \quad (1)$$

Similarly, the entity e is characterized by the entity name and its textual description.

$$e = \text{[CLS] name [ENT] description [SEP]} \quad (2)$$

where [CLS], [Ms], [Me], [ENT] and [SEP] are special tokens to mark the boundaries of the different pieces of information. Let the mention $m = \{\mathbf{x}^m_t\}^{|m|}_{t=1}$ with $|m|$ wordpieces, and the entity description $e = \{\mathbf{x}^e_{t'}\}^{|e|}_{t'=1}$ with $|e|$ wordpieces. We extract the corresponding representations $H_m \in R^{d\times|m|}$ and $H_e \in R^{d\times|e|}$ as follows:

$$\begin{aligned} H_m &= E_{\theta_m}(m) \\ H_e &= E_{\theta_e}(e) \end{aligned} \quad (3)$$

where d denotes the dimension of representations.

Then, the entity linking problem is reduced to quantifying the similarity between H_m and H_e using a score function f, i.e., $f(H_m, H_e)$. If the mention-entity pair (m, e) matches, the score $f(H_m, H_e)$ should be high, or low if otherwise. Wu et al. (2020) defines a DUAL score function that takes the [CLS] representations $h^m_{\text{[CLS]}} \in R^{d\times 1}$ and $h^e_{\text{[CLS]}} \in R^{d\times 1}$ of the respective representations H_m and H_e to compute the score $f(H_m, H_e)$.

DUAL:

$$f(H_m, H_e) = (h^m_{\text{[CLS]}})^T h^e_{\text{[CLS]}} \quad (4)$$

Recently, Zhang and Stratos (2021) followed the architecture of Wu et al. (2020) and showed that the Sum-Of-Max (SOM) scorer (Khattab and Zaharia, 2020) yields better results in comparison to DUAL. SOM computes $f(H_m, H_e)$ as follows.

SOM:

$$f(H_m, H_e) = \sum_{t=1}^{|m|} \max_{t'=1}^{|e|} (h^m_t)^T h^e_{t'} \quad (5)$$

However, it is worth to note that the SOM scorer comes at the expense of increased computational cost due to the consideration of all hidden states of H_m and H_e in the scorer.

Finally, the model is trained to encourage discrimination between golden mention-entity pairs and negative mention-entity pairs. We minimize a standard loss function L based on the empirical estimate of the NCE loss (Gutmann and Hyvärinen, 2010),

$$L(\theta_m, \theta_e) = -\frac{1}{M}\sum_{i=1}^{M} \log \frac{\exp(f(E_{\theta_m}(m_i), E_{\theta_e}(e_{i,1})))}{\sum_{j=1}^{C} \exp(f(E_{\theta_m}(m_i), E_{\theta_e}(e_{i,j})))} \quad (6)$$

where $\{(m_i, e_{i,1})\}_{i=1,...,M}$ are golden mention-entity pairs in the training set, and $\{e_{i,2}, ..., e_{i,C}\}$ are $C-1$ negative entities for the i-th mention.

Regarding model training, two main strategies have been considered in previous work to construct negative mention-entity pairs: (1) **Random Cross-Domain (Random-CD)**: negative entities for a given mention are randomly sampled from the entity set. (2) **Hard Cross-Domain (Hard-CD)**: In a training epoch, all entities are first ranked with the current trained model and the top-k entities are taken as hard negative entities. Both strategies sample negatives across all domains. While Random-CD aim to select negative entities that are semantically different from the golden entity, Hard-CD additionally aims to select negatives close to the golden entity in the representation space.

3.3 Transformational Biencoder

We found that the Biencoder assumes that leveraging the common knowledge in BERT is sufficient to achieve zero-shot transfer from source to target domain. However, while fine-tuned BERT in-domain can achieve state-of-the-art performance, its zero-shot performance on the target domain can deteriorate substantially (Ma et al., 2019). As a solution, we add a learnable transformation into the Biencoder to achieve zero-shot transfer in the training process. Otherwise, the training proceeds in the standard way and learns the parameters of E_{θ_m} and E_{θ_e}. We refer to this modified Biencoder as an Transformational Biencoder (T-Biencoder).

We present the architecture of T-Biencoder in Figure 2. $E_{\theta_m^1}$ and $E_{\theta_m^2}$ are the respective early and later transformer layers of the BERT architecture E_{θ_m}, where $\theta_m = \{\theta_m^1, \theta_m^2\}$. We use the parallel notations $E_{\theta_e^1}$ and $E_{\theta_e^2}$ for the BERT architecture E_{θ_e}, where $\theta_m = \{\theta_e^1, \theta_e^2\}$. The encoders $E_{\theta_m^1}$ and $E_{\theta_e^1}$ aim to map the respective mention m and entity e into a common space Z. Since the common space Z that best fit our model is unknown, the number of layers of $E_{\theta_m^1}$ and $E_{\theta_e^1}$ is taken as a hyper-parameter K. This is also the K-th layer of E_{θ_m} and E_{θ_e}. Now, Z_m and Z_e, the respective representations of m and e in the common space Z are computed as follows:

$$\begin{aligned} Z_m &= E_{\theta_m^1}(m) \\ Z_e &= E_{\theta_e^1}(e) \end{aligned} \tag{7}$$

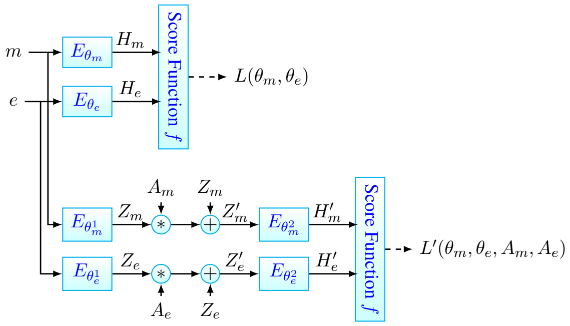


Figure 2: Architecture of T-Biencoder

where $Z_m \in R^{d\times|m|}$ and $Z_e \in R^{d\times|e|}$. In the common space we assume relatedness between source and target domains to achieve zero-shot transfer. Hence, there exist at least one transformation $\bar{A}_m$ (or $\bar{A}_e$) to transform the mention (or entity) pair in the source domain distribution $\mathcal{S}$ to the target domain distribution $\mathcal{T}$. Let A_m and A_e be the learnable transformation matrices which aims to approximate $\bar{A}_m$ and $\bar{A}_e$ respectively. The representations $Z_m \sim \mathcal{S}$ and $Z_e \sim \mathcal{S}$ are transformed into the representations $Z'_m \sim \mathcal{T}$ and $Z'_e \sim \mathcal{T}$:

$$\begin{aligned} Z'_m &= Z_m + A_m Z_m \\ Z'_e &= Z_e + A_e Z_e \end{aligned} \tag{8}$$

The final representations H'_m and H'_e are then constructed by feeding Z'_m and Z'_e into the encoders $E_{\theta_m^2}$ and $E_{\theta_e^2}$.

$$\begin{aligned} H'_m &= E_{\theta_m^2}(Z'_m) \\ H'_e &= E_{\theta_e^2}(Z'_e) \end{aligned} \tag{9}$$

Finally, we feed H'_m and H'_e into the score function f and calculate the loss $L'(\theta_m, \theta_e, A_m, A_e)$. The total loss $\mathcal{L}_{\text{total}}$ of our model is formulated as

$$\begin{aligned} \mathcal{L}_{\text{total}} = \min_{\theta_m, \theta_e} [L(\theta_m, \theta_e) + \\ \max_{||A_m||, ||A_e|| \leq \epsilon} L'(\theta_m, \theta_e, A_m, A_e)] \end{aligned} \tag{10}$$

where L is the standard loss function. L' is a transformational loss function which follows the same definition as L. The hyper-parameter ϵ quantifies the supremum of shift between source and target distributions. Besides learning θ_m and θ_e, the second term in $\mathcal{L}_{\text{total}}$ aims to find transformations A_m and A_e that maximizes L' conditioned by $||A_m||, ||A_e|| \leq \epsilon$. Specifically, we find the worst A_m and A_e such that $L'(\theta_m, \theta_e, \bar{A}_m, \bar{A}_e) \leq L'(\theta_m, \theta_e, A_m, A_e)$ given ϵ. The idea is that, if the encoders E_{θ_m} and E_{θ_e} can work for representations obtained through A_m and A_e, they can also work for representations obtained through the ground-truth transformations $\bar{A}_m$ and $\bar{A}_e$. Note, the transformations only serve to shift the source to the target distribution. Hence, at inference time the entities and mentions are fed directly to the encoders E_{θ_e} and E_{θ_m} without being transformed by A_m and A_e. We experiment with the DUAL and SOM scorers in our work.

3.4 Negative Sampling In-Domain

We point out that carefully constructing negatives is crucial to performance. By leveraging Hard-CD, Zhang and Stratos (2021) improves upon the task, indicating its benefit over Random-CD.

While these sampling strategies have shown its benefit, they disregard the domain of the golden entity. We posit that negatives that are lexically similar to the golden entity should be additionally considered to obtain harder negatives. That is, negatives that are lexically similar, semantically different and close to the representation of the golden entity are harder. We therefore propose two negative samplig strategies: (1) **Random In-Domain (Random-ID)**: randomly samples negative entities in-domain of the golden entity. (2) **Hard In-Domain (Hard-ID)**: all entities in-domain of golden entity are ranked in a training epoch, and the top-k entities are taken as hard negatives. We will demonstrate through extensive experiments to show the effectiveness of negative sampling in-domain.

4 Experiments

4.1 Dataset

We follow the recent works (Logeswaran et al., 2019; Wu et al., 2020; Tang et al., 2021; Zhang and Stratos, 2021) and evaluate on the Zeshel dataset (Logeswaran et al., 2019),[1] which is a prevailing benchmark dataset for zero-shot entity linking. Zeshel contains 16 specialized domains from Wikia,[2] partitioned into 8 domains for training, and 4 domains each for validation and testing. Table 4 shows the dataset statistics, including the number of entities and mentions.

4.2 Evaluation Protocol

EL systems typically follow a two-stage pipeline: (1) a candidate generation stage, where an entity retriever is trained to select top-k candidates for each given mention, (2) a candidate ranking stage, where a ranker is trained to identify the golden entity among selected candidates for a given mention. The candidate generation is essential to the performance of candidate ranking because if the golden entity is not retrieved in the top-k candidates, the model can never recover the golden entity during candidate ranking. We therefore follow the evaluation protocol of previous work (Logeswaran et al., 2019; Wu et al., 2020; Zhang and Stratos, 2021) and evaluate at the candidate generation stage. We report micro-averaged top-64 recall for models on the validation set and testing set. Thus, we consider a mention's prediction to be correct if its golden entity is included in the top-64 candidates. Average results over 3 runs are reported for our models.

Domains	Entities	Mentions	
		Train	Evaluation
Training			
American Football	31929	3898	743
Doctor Who	40281	8334	1521
Fallout	16992	3286	593
Final Fantasy	14044	6041	1156
Military	104520	13063	2764
Pro Wrestling	10133	1392	262
Star Wars	87056	11824	2706
World of Warcraft	27677	1437	255
Validation			
Coronation Street	17809	0	1464
Muppets	21344	0	2028
Ice Hockey	28684	0	2233
Elder Scrolls	21712	0	4275
Testing			
Forgotten Realms	15603	0	1200
Lego	10076	0	1199
Star Trek	34430	0	4227
YuGiOh	10031	0	3374

Table 1: Statistic of the Zeshel dataset.

4.3 Implementation Details

To fairly compare with recent work (Wu et al., 2020), we use the BERT-base-uncased (Devlin et al., 2018) as the text encoder, where the embedding layer is kept frozen and the layers are fine-tuned during training. We directly use the preprocessed dataset provided by (Wu et al., 2020),[3] where mentions/entities are represented by $n = 128$ wordpiece tokens, i.e., $n = |m| = |e|$. Number of negative entities for each mention is 15. We employ AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate ($lr = 1e^{-5}$) warm-up schedule to smoothen the training process. We train

[1] https://github.com/lajanugen/zeshel

[2] 3 https://www.wikia.com

[3] https://github.com/facebookresearch/BLINK

Model	Scorer	Negatives	Evaluation		Time
			Validation	Testing	
(Logeswaran et al., 2019)	BM25	-	76.22	69.13	-
Biencoder (Wu et al., 2020)	DUAL	Random-CD	91.44	82.06	12
Biencoder (Zhang and Stratos, 2021)	DUAL	Random-CD	91.08	81.80	12
Biencoder (Zhang and Stratos, 2021)	DUAL	Hard-CD	91.99	84.87	110
Biencoder (Zhang and Stratos, 2021)	SOM	Random-CD	92.51	87.62	13
Biencoder (Zhang and Stratos, 2021)	SOM	Hard-CD	94.66	89.62	2306
T-Biencoder (Ours)	DUAL	Random-ID	92.43±0.19	85.72±0.11	21
	DUAL	Hard-ID	**93.03±0.10**	**86.35±0.15**	131
	SOM	Random-ID	94.56±0.32	90.68±0.21	23
	SOM	Hard-ID	**95.49±0.23**	**91.16±0.14**	625

Table 2: Results of compared models are retrieved from their original papers. Models' efficiency is the time cost per epoch in minutes. Results of our models are the average over 3 runs using different random seeds.

our models for 5 epochs using a batch size of 64 for mention-entity pairs. We perform a grid search to select the best set of hyper-parameters: ϵ in $[1e^{-3}, 1e^{-2}, 1e^{-1}, 1.0, 10, 20, \ldots, 100]$ and K in $[0, 1, 4, 8]$. Best hyperparameter values are shown in Table 3. All models are trained in parallel on four NVIDIA V100 32GB.

Scorer	Negative	ϵ	K
DUAL	Random-ID	32	1
DUAL	Hard-ID	42	1
SOM	Random-ID	20	1
SOM	Hard-ID	40	1

Table 3: Best observed hyper-parameter configurations of T-Biencoder on the validation set.

4.4 Performance Comparison

In this section we compare our model against recent work (Wu et al., 2020; Zhang and Stratos, 2021) for candidate generation. These works employ the Biencoder and generate negative entities for optimization. DUAL and SOM scorers are used in this work. As a baseline we include the work by Logeswaran et al. (2019) which uses the BM25. Note, the following works (Tang et al., 2021; Yao et al., 2020) are excluded since they evaluate at the candidate ranking stage, making their results uncomparable with ours.

4.4.1 Main Results

Results of compared models are shown in Table 2. BM25 shows poor performance due to the emphasis on lexical similarity between mention and candidate entity tokens. In contrast, methods (Wu et al., 2020; Zhang and Stratos, 2021) that generate semantic representations have shown to be effective, with an improvement of at least 12.93% on the test set. First, we compare the performance of these models with respect to the DUAL scorer on the test set. We find that T-Biencoder outperforms the Biencoder (Zhang and Stratos, 2021) by 3.92% for random sampling, and by 1.48% for hard sampling. With respect to the SOM scorer, we observe that T-Biencoder outperforms Biencoder by 3.06% for random sampling, and by 1.54% for hard sampling. These results indicate the effectiveness of our sampling strategies (i.e., Random-ID and Hard-ID) as well as the transformation approach. We also find that SOM yields better results over DUAL while hard sampling leads to better optimization. However, leveraging SOM and hard sampling increases computational cost by orders of magnitude, as we examine its efficiency in Section 4.4.3.

Interestingly, we find that by using Random-ID for either SOM or DUAL, we achieve better performance in comparison to Hard-CD, indicating the effectiveness and efficiency of our model. T-Biencoder achieves state-of-the-art results for either DUAL and SOM scorers, and additionally shows stability given the low standard deviations.

4.4.2 Domain Zero-Shot Performance

Our main results show that we achieve state-of-the-art on both validation and testing sets. To show that this improvement is true for all validation/test domains and not as a result of a specific validation/test domain, we show more fine-grained results. Specifically, we report the domain zero-shot performance for Biencoder and T-Biencoder using the Random-ID sampling strategy. Table 4 shows the results for the different validation and testing domains.

Due to the degree of dissimilarity between the train and validation/test domains, naive zero-shot transfer by the Biencoder produces unsatisfactory results. By learning the transformation, T-Biencoder outperforms Biencoder. We found that the test domain "YuGiOh" is closely related to the train domains "Star Wars" and "Final Fantansy" in the sense that they belong to the super-domain of comics. By exploiting the relatedness of these domains, we achieve impressive results on “YuGiOh”. Specifically, “YuGiOh” improves by 2.37% for DUAL and 2.48% for SOM using the T-Biencoder.

Domains	Biencoder		T-Biencoder	
	DUAL	SOM	DUAL	SOM
Validation				
Coronation Street	88.87	92.28	**89.07**	**92.76**
Muppets	88.51	91.62	**89.89**	**92.75**
Ice Hockey	92.03	92.07	**92.39**	**93.42**
Elder Scrolls	94.32	96.05	**94.85**	**96.28**
Testing				
Forgotten Realms	93.83	97.00	**94.25**	**97.17**
Lego	92.16	94.50	**92.83**	**95.58**
Star Trek	85.78	89.05	**86.56**	**89.64**
YuGiOh	77.12	85.66	**79.49**	**88.14**

Table 4: Zero-shot Performance on Different Domains under Random-ID

4.4.3 Efficiency Analysis

We dive into the efficiency of compared models. Table 2 shows the cost per epoch, measured in minutes (last column). In spite of the performance advantages of using SOM and hard sampling, the computational and memory requirements is expensive. We draw this conclusion from the complexity of the SOM scorer and the mechanism behind hard sampling.

Given a mention-entity (m, e) pair in the train set, where n is the length of the mention/entity. DUAL has a complexity $\mathcal{O}(1)$ (see (4)) while SOM has a complexity $\mathcal{O}(n^2)$ (see (5)) to analyze (m, e). This means DUAL is bounded by a constant while SOM scales quadratically in computation and memory storage with the length of the mention or entity. On the other hand, hard sampling requires that the set of entities are ranked by the scorer before selecting the top-k negative entities. Random sampling requires no scorer. Taking these two factors (i.e., scorer’s complexity and sampling technique) into consideration, it is not surprising to see (1) the time cost of hard sampling to be significantly higher than random sampling, (2) the time cost of SOM with hard sampling to be larger than that of DUAL with hard sampling. However, it is interesting to see that T-Biencoder with Hard-ID trains about 3.7 (2306/625) times faster than the Biencoder with Hard-CD for the SOM scorer. We attribute this efficiency to sampling in-domain, where the in-domain’s entity set is significantly smaller than the entity set of all domains in the train set.

4.5 Ablation Study

Given the difference between the Biencoder (Zhang and Stratos, 2021) and T-Biencoder, our performance can be attributed to the learned transformation or/and our negative sampling strategy. In this section, we investigate the contribution of the different model components through ablation studies. We experiment with the negative sampling strategies using the DUAL and SOM scorers. Due to the computational cost constraints, we do not report results for SOM using Hard strategies. Table 5 shows the results of this experiment.

Model	**Scorer**	**Negatives**	**Evaluation**	
			Validation	Testing
Biencoder	DUAL	Random-CD	91.03	81.88
Biencoder	DUAL	Random-ID	92.08	84.82
T-Biencoder	DUAL	Random-ID	**92.43**	**85.72**
Biencoder	DUAL	Hard-CD	92.03	84.97
Biencoder	DUAL	Hard-ID	92.70	85.50
T-Biencoder	DUAL	Hard-ID	**93.03**	**86.35**
Biencoder	SOM	Random-CD	92.18	87.82
Biencoder	SOM	Random-ID	93.81	89.46
T-Biencoder	SOM	Random-ID	**94.56**	**90.68**

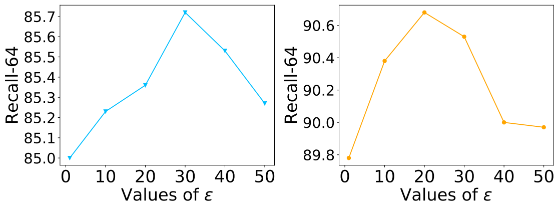
Table 5: Ablation Study: Model Performance by sampling negatives in-domain or across-domains of the golden entity.

We fix the negative sampling used, in order to determine the contribution of the learned transformation. Using Random-ID (or Hard-ID) and DUAL, we find that T-Biencoder outperforms the Biencoder by 0.9% (or 0.85%) on the test set. Using Random-ID and SOM, we find that T-Biencoder outperforms the Biencoder by 1.22% on the test set. We see similar performance situations on the validation set. These results demonstrate that the learned transformation leads to improvement irrespective of the score function or negative sampling strategy used.

We also demonstrate the effectiveness of in-domain sampling by comparing against cross-domain sampling. With the DUAL scorer, we find that the Biencoder achieves gains of 2.94% (or 0.53%) on the test set with Random-ID (or Hard-ID) as against using Random-CD (or Hard-CD). With the SOM scorer, we find that the Biencoder achieves gains of 1.64% on the test set with Random-ID as against using Random-CD. We see similar performance situations on the validation set. These results indicate that our in-domain negative sampling strategy produces harder negatives, resulting in strong gradient signals for optimization. By leveraging both in-domain negative sampling as well as transformation learning, we achieve state-of-the-art performance.

4.6 Impact of ϵ and K

In this section, we investigate the sensitivity of T-Biencoder in terms of the hyper-parameters ϵ and K. Recall, K indicates the layer in which we apply a transformation on its output during training. Suppose $K = 0$, a transformation is applied on the output of the embedding layer. ϵ on the other hand represents the upper bound for the norm of transformation in (10). We perform this experiment on the test set under Random-ID using our best hyper-parameter values. Figure 3 shows the Recall-64 curve for different values of ϵ, with $K = 1$. Table 6 also shows the performance for different values of K, with $\epsilon = 32$ on DUAL and $\epsilon = 20$ on SOM.



(A) DUAL (B) SOM

Figure 3: Recall-64 curves for different ϵ under Random-ID for T-Biencoder

.

Scorer	$K=0$	$K=1$	$K=4$	$K=8$
DUAL	85.45	85.72	85.12	84.36
SOM	90.23	90.68	90.05	88.69

Table 6: Recall-64 performance for different K under Random-ID for T-Biencoder.

In Table 6, we find that DUAL achieves the best performance in the early layers, i.e., $K = 1$. Meanwhile, with increasing K, the performance deteriorates. Since representations for source and target domains tend to share a low-level linguistic representation space in early transformer layers while reserving higher layers for the task or domain specific knowledge (Jawahar et al., 2019; Durrani et al., 2021), we believe a transformation easily bridges the domain gap in such layers to achieve performance. With regard to the impact of ϵ, Figure 3 shows that the performance of T-Biencoder increases with increasing values of ϵ up to a certain point, achieving scores of 85.72 on DUAL and 90.68 on SOM. After this point the performance becomes unstable and deteriorates, indicating the importance of controlling ϵ.

4.7 Analyzing the Importance of L

The loss functions L and L' both tune the parameters of the encoder. But L' additionally focuses on mitigating the domain shift problem at the same time. In this section, we wish to answer the question: Can we produce good mention-entity representations through only L'? To answer this question, we ablate T-Biencoder by removing the standard loss L, constructing the ablated model T-Biencoder$_{\text{wo L}}$. We consider both DUAL and SOM under the Random-ID strategy. We use the best hyper-parameter values for ablated models. Table 7 shows the results of our experiments. The last two columns is the performance on the held-out mentions in the training set. (see Table 4). We find that the ablated model deteriorates for both DUAL and SOM, indicating that we cannot obtain good sentence representations by minimizing only L'.

Model	Scorer	Testing	Training	
			Seen	Unseen
Biencoder	DUAL	84.82	95.65	94.97
T-Biencoder	DUAL	85.72	96.07	95.52
T-Biencoder$_{\text{wo L}}$	DUAL	82.11	91.97	91.52
Biencoder	SOM	89.46	96.30	96.12
T-Biencoder	SOM	90.68	96.68	96.56
T-Biencoder$_{\text{wo L}}$	SOM	86.68	94.64	94.38

Table 7: Importance of L: Performance of different models, including the ablated model T-Biencoder$_{\text{wo L}}$.

5 Conclusion

We introduced a Transformational Biencoder (T-Biencoder) that builds upon the recent proposed Biencoder (Wu et al., 2020) to solve the problem of zero-shot entity linking. Our work shows how to explicitly improve the zero-shot transfer capability of the Biencoder for EL. We hypothesized that negative samples drawn in-domain of the golden-entity results in better optimization. Our experimental analysis demonstrates that this assumption holds, where we see the benefits of our negative sampling strategy as well as the learned transformation. Our results show strong improvements for the task, in both effectiveness and efficiency. We envision that the T-Biencoder can further be improved by learning a transformation for each domain in the train set, since a single transformation may not effectively capture the relationship among different specialized domains. We leave this for future work.

Acknowledgements

This work is supported partly by the Fundamental Research Funds for the Central Universities, by the State Key Laboratory of Software Development Environment, and by a Leverhulme Trust Research Project Grant.

References

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. 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.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285.

Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. 2021. How transfer learning impacts linguistic knowledge in deep NLP models? In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4947–4957. Association for Computational Linguistics.

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

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.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? 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*, pages 3651–3657. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Thomas Lin, Oren Etzioni, et al. 2012. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 84–88.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3449–3460. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with bert-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Avirup Sil, Ernest Cronin, Penghai Nie, Yinfei Yang, Ana-Maria Popescu, and Alexander Yates. 2012. Linking named entities to any database. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 116–127. ACL.

Hongyin Tang, Xingwu Sun, Beihong Jin, and Fuzheng Zhang. 2021. A bidirectional multi-paragraph reading model for zero-shot entity linking. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13889–13897. AAAI Press.

Han Wang, Jinguang Zheng, Xiaogang Ma, Peter Fox, and Heng Ji. 2015. Language and domain independent entity linking with quantified collective validation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 695–704. The Association for Computational Linguistics.

Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270.

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 2020, Online, November 16-20, 2020*, pages 6397–6407. Association for Computational Linguistics.

Zonghai Yao, Liangliang Cao, and Huapu Pan. 2020. Zero-shot entity linking with efficient long range sequence modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2517–2522. Association for Computational Linguistics.

Wenzheng Zhang and Karl Stratos. 2021. Understanding hard negatives in noise contrastive estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1090–1101. Association for Computational Linguistics.