Contextualized End-to-End Neural Entity Linking

Haotian Chen
BlackRock
haotian.chen@blackrock.com

Andrej Zukov-Gregoric
BlackRock
andrej.zukovgregoric@blackrock.com

Xi (David) Li
BlackRock
david.li@blackrock.com

Sahil Wadhwa
University of Illinois at Urbana-Champaign*
sahilw2@illinois.edu

Abstract

We propose an entity linking (EL) model that jointly learns mention detection (MD) and entity disambiguation (ED). Our model applies task-specific heads on top of shared BERT contextualized embeddings. We achieve state-of-the-art results across a standard EL dataset using our model; we also study our model's performance under the setting when hand-crafted entity candidate sets are not available and find that the model performs well under such a setting also.

1 Introduction

Entity linking (EL)[1], in our context, refers to the joint task of recognizing named entity mentions in text through mention detection (MD) and linking each mention to a unique entity in a knowledge base (KB) through entity disambiguation (ED)[2]. For example, in the sentence *"The Times began publication under its current name in 1788,"* the span *The Times* should be detected as a named entity mention and then linked to the corresponding entity: *The_Times*, a British newspaper. However, an EL model which disjointly applies MD and ED might easily mistake this mention with *The_New_York_Times*, an American newspaper. Since our model jointly learns MD and ED from the same contextualized BERT embeddings, its final EL prediction is partially informed by both. As a result, it is able to generalize better.

Another common approach employed in previous EL research is candidate generation, where for each detected mention, a set of candidate entities is generated and the entities within it are ranked by a model to find the best match. Such sets are built using hand-crafted rules which define which entities make it in and which do not. This risks (1) skipping out on valid entities which should be in the candidate set and (2) inflating model performance since often times candidate sets contain only one or two items. These sets are almost always used at prediction time and sometimes even during training. Our model has the option of not relying on them during prediction, and never uses them during training.

We introduce two main contributions:

(i) We propose a new end-to-end differentiable neural EL model that jointly performs MD and ED and achieves state-of-the-art performance.

(ii) We study the performance of our model when candidate sets are removed to see whether EL can perform well without them.

2 Related Work

Neural-network based models have recently achieved strong results across standard EL datasets. Research has focused on learning better entity representations and extracting better local and global features through novel model architectures.

Entity representation. Good KB entity representations are a key component of most ED and EL models. Representation learning has been addressed by Yamada et al. (2016), Ganea and Hofmann (2017), Cao et al. (2017) and Yamada et al. (2017). Sil et al. (2018) and Cao et al. (2018) extend it to the cross-lingual setting. More recently, Yamada and Shindo (2019) have suggested learning entity representations using BERT which achieves state-of-the-art results in ED.

Entity Disambiguation (ED). The ED task assumes already-labelled mention spans which are then disambiguated. Recent work on ED has focused on extracting global features (Ratinov et al.,

*Work done while at BlackRock.

[1]Also known as A2KB task in GERBIL evaluation platform (Röder et al., 2018) and end-to-end entity linking in some literature

[2]Also known as D2KB task in GERBIL

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 637–642

2011; Globerson et al., 2016; Ganea and Hofmann, 2017; Le and Titov, 2018), extending the scope of ED to more non-standard datasets (Eshel et al., 2017), and positing the problem in new ways such as building separate classifiers for KB entities (Barrena et al., 2018).

Entity Linking (EL). Early work by Sil and Yates (2013), Luo et al. (2015) and Nguyen et al. (2016) introduced models that jointly learn NER and ED using engineered features. More recently, Kolitsas et al. (2018) propose a neural model that first generates all combinations of spans as potential mentions and then learns similarity scores over their entity candidates. MD is handled implicitly by only considering mention spans which have non-empty candidate entity sets. Martins et al. (2019) propose training a multi-task NER and ED objective using a Stack-LSTM (Dyer et al., 2015). Finally, Poerner et al. (2019) and Broscheit (2019) both propose end-to-end EL models based on BERT. Poerner et al. (2019) model the similarity between entity embeddings and contextualized word embeddings by mapping the former onto the latter whereas Broscheit (2019) in essence do the opposite. Our work is different in three important ways: our training objective is different in that we explicitly model MD; we analyze the performance of our model when candidate sets are expanded to include the entire universe of entity embeddings; and we outperform both models by a wide margin.

3 Model Description

Given a document containing a sequence of n tokens $\mathbf{w} = \{w_1, ..., w_n\}$ with mention label indicators[3] $\mathbf{y}_{md} = \{I, O, B\}^n$ and entity IDs $\mathbf{y}_{ed} = \{j \in \mathbb{Z} : j \in [1, k]\}^n$ which index a pre-trained entity embedding matrix $\mathbf{E} \in \mathbb{R}^{k \times d}$ of entity universe size k and entity embedding dimension d, the model is trained to tag each token with its correct mention indicator and link each mention with its correct entity ID.

3.1 Text Encoder

The text input to our model is encoded by BERT (Devlin et al., 2019). We initialize the pre-trained weights from BERT-BASE.[4] The text input is tokenized by the cased WordPiece (Johnson et al., 2017) sub-word tokenizer. The text encoder outputs n contextualized WordPiece embeddings $\mathbf{h}$ which are grouped to form the embedding matrix $\mathbf{H} \in \mathbb{R}^{n \times m}$, where m is the embedding dimension. In the case of BERT-BASE, m is equal to 768.

The transformation from word level to WordPiece sub-word level labels is handled similarly to the BERT NER task, where the head WordPiece token represents the entire word, disregarding tail tokens.

BERT comes in two settings: feature-based and fine-tuned. Under the feature-based setting, BERT parameters are not trainable in the domain task (EL), whereas the fine-tuned setting allows BERT parameters to adapt to the domain task.

3.2 EL model

MD is modeled as a sequence labelling task. Contextualized embeddings $\mathbf{h}$ are passed through a feed-forward neural network and then softmaxed for classification over IOB:

$$\mathbf{m}_{md} = \mathbf{W}_{md}\mathbf{h} + \mathbf{b}_{md} \tag{1}$$

$$\mathbf{p}_{md} = \text{softmax}(\mathbf{m}_{md}) \tag{2}$$

where $\mathbf{b}_{md} \in \mathbb{R}^3$ is the bias term, $\mathbf{W}_{md} \in \mathbb{R}^{3 \times m}$ is a weight matrix, and $\mathbf{p}_{md} \in \mathbb{R}^3$ is the predicted distribution across the $\{I, O, B\}$ tag set. The predicted tag is then simply:

$$\hat{\mathbf{y}}_{md} = \arg\max_i \{\mathbf{p}_{md}(i)\} \tag{3}$$

ED is modeled by finding the entity (during inference this can be from either the entire entity universe or some candidate set) closest to the predicted entity embedding. We do this by applying an additional ED-specific feed-forward neural network to $\mathbf{h}$:

$$\begin{aligned} \mathbf{m}_{ed} &= \tanh(\mathbf{W}_{ed}\mathbf{h} + \mathbf{b}_{ed}) \\ \mathbf{p}_{ed} &= s(\mathbf{m}_{ed}, \mathbf{E}) \\ \hat{\mathbf{y}}_{ed} &= \arg\max_j \{\mathbf{p}_{ed}(j)\} \end{aligned} \tag{4}$$

where $\mathbf{b}_{ed} \in \mathbb{R}^d$ is the bias term, $\mathbf{W}_{ed} \in \mathbb{R}^{d \times m}$ is a weight matrix, and $\mathbf{m}_{ed} \in \mathbb{R}^d$ is the same size as the entity embedding and s is any similarity measure which relates $\mathbf{m}_{ed}$ to every entity embedding in $\mathbf{E}$. In our case, we use cosine similarity. Our

[3] We use standard *inside-outside-beginning* (IOB) tagging format introduced by (Ramshaw and Marcus, 1995)

[4] https://github.com/google-research/bert

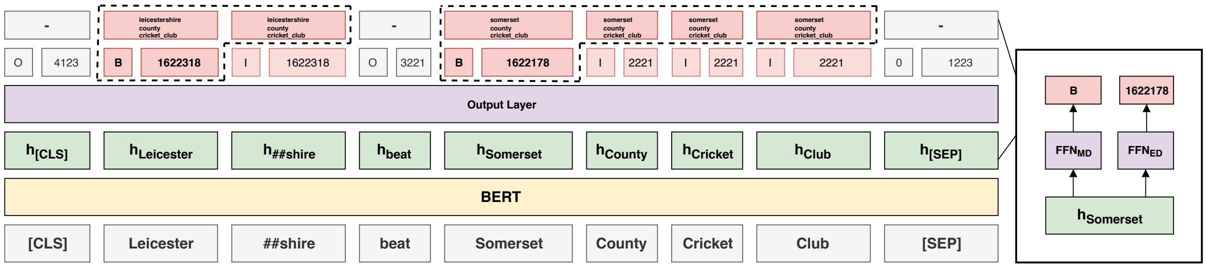

Figure 1: Architecture of the proposed model. WordPiece tokens are passed through `BERT` forming contextualized embeddings. Each contextualized embedding is passed through two task-specific feed-forward neural networks for MD and ED, respectively. Entity ID prediction on the 'B' MD tag is extended to the entire mention span.

predicted entity is the index of $\mathbf{p}_{ed}$ with the highest similarity score.

We use pre-trained entity embeddings from *wikipedia2vec* (Yamada et al., 2018), as pre-training optimal entity representation is beyond the scope of this work. Ideally, pre-trained entity embeddings should be from a similar architecture to our EL model, but experiments show strong results even if they are not. The *wikipedia2vec* entity embeddings used in our model are trained on the 2018 Wikipedia with 100 dimensions and link graph support.[5]

During inference, after receiving results for each token from both the MD and ED tasks, the mention spans are tagged with $\{B, I\}$ tags as shown in Figure 1. For each mention span, the entity ID prediction of first token represents the entire mention span. The remaining non-mention and non-first entity ID prediction are masked out. Such behavior is facilitated by the training objective below.

During training, we minimize the following multi-task objective which is inspired by Redmon and Farhadi (2017) from the object detection domain:[6]

$$J(\theta) = \lambda \mathcal{L}_{md}(\theta) + (1 - \lambda)\mathcal{L}_{ed}(\theta) \qquad (5)$$

where $\mathcal{L}_{md}$ is the cross entropy between predicted and actual distributions of IOB and $\mathcal{L}_{ed}$ is the cosine similarity between projected entity embeddings and actual entity embeddings. We tentatively explored triplet loss and contrastive loss with some simple negative mining strategies for ED but did not observe significant gains in performance. The two loss functions are weighted by a hyperparameter λ (in our case $\lambda = 0.1$). Note that $\mathcal{L}_{md}$ is calculated for all non-pad head WordPiece tokens but $\mathcal{L}_{ed}$ is calculated only for the first WordPiece token of every labeled entity mention with a linkable and valid entity ID label.

[5] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

[6] Similar to EL, object detection has two sub-tasks: locating bounding boxes and identifying objects in each box.

4 Experiments

4.1 Dataset and Performance Metrics

We train and evaluate our model on the widely used AIDA/CoNLL dataset (Hoffart et al., 2011). It is a collection of news articles from Thomson Reuters, which is split into training, validation (testa) and test (testb) sets. Following convention, the evaluation metric is strong-matching span-level InKB micro and macro F1 score over gold mentions, where entity annotation is available (Röder et al., 2018). Note that ED models are evaluated by accuracy metric while EL models are evaluated by F1, which penalizes the tagging of non-mention spans as entity mentions.

4.2 Candidate Sets

All EL models cited rely on candidate sets. As for our model, mentions can be efficiently disambiguated with respect to the entire entity universe, which we take to be the one million most frequent entities in 2018 Wikipedia. Consequently, our model can circumvent candidate generation, as well as the external knowledge that comes with it. In order to study the impact of candidate sets on our model, we apply candidate sets from Hoffart et al. (2011) backed by the YAGO knowledge graph (Suchanek et al., 2007). Importantly, we do not arbitrarily limit the size of the candidate sets.

4.3 Training Details and Settings

We train the EL model on the training split with a batch size of 4 for 50,000 steps. As in the original `BERT` paper, the model is optimized by the Adam optimizer (Kingma and Ba, 2014) with the same hyperparameters except the learning rate, which we set to be 2e-5. Training was performed on a Tesla V100 GPU. A 0.1 dropout rate was used on the prediction heads. Experiments are repeated three times to calculate an error range.

4.4 Results

Comparison with Other EL Models. We compare our model with six of the most recent, and best performing, EL models in Table 1. We study the performance of our model with, and without candidate sets (see Section 4.2). We find that when candidate sets are provided, our model outperforms existing models by a significant margin.

One of the problems of comparing results in the EL and ED space is that candidate sets are usually paper-specific and many works suggest their own methodologies for generating them. In addition to using candidate sets from Hoffart et al. (2011) (which makes us comparable to Kolitsas et al. (2018) who use the same sets), we impose no arbitrary limit on candidate set size. This means that many of our candidate sets have more than the standard 20-30 candidates, which are normally considered in past works.

Without candidate sets our model also shows good results and validation performance is on par with recent work by Martins et al. (2019) who used stack LSTMs *with* candidate sets. We improve upon work by Broscheit (2019) who, like us, do not use candidate sets. We use a larger overall entity universe (1M instead of 700K). Interestingly, Broscheit (2019) note that during their error analysis only 3% of wrong predictions were due to erroneous span detection. This could potentially explain our margin of improvement in the test set since our model is span-aware unlike theirs. For more details on the properties of the AIDA dataset we recommend Ilievski et al. (2018).

Overfitting. There are considerable drops in performance between validation and test both when BERT is fine-tuned or fixed, pointing to potential problems with overfitting. Identical behaviour is seen in Broscheit (2019) and Poerner et al. (2019), who propose similar BERT-based models. Whether overfitting is due to BERT or the downstream models requires further research.

Even more considerable drops in performance between validation and test are experienced when candidates sets are not used and entities are linked across the entire entity universe. We cannot be sure whether these drops are specific to BERT since no non-BERT works cite results over the entire entity universe.

	AIDA/testa F1 (val)		AIDA/testb F1 (test)	
	Macro	Micro	Macro	Micro
Martins et al. (2019)	82.8	85.2	81.2	81.9
Kolitsas et al. (2018)	86.6	89.4	82.6	82.4
Cao et al. (2018)	77.0	79.0	80.0	80.0
Nguyen et al. (2016)	-	-	-	78.7
Broscheit (2019)	-	76.5	-	67.8
Poerner et al. (2019)	89.1	90.8	84.2	85.0
Fine-tuned BERT with candidate sets	**92.6**±0.2	**93.6**±0.2	**87.5**±0.3	**87.7**±0.3
Fine-tuned BERT without candidate sets	82.6±0.2	83.5±0.2	70.7±0.3	69.4±0.3

Table 1: Strong-matching span-level InKB macro & micro F1 results on validation and test splits of AIDA/CoNLL dataset. Note that the other models cited all use candidate sets. We run our models three times with different seeds to get bounds around our results.

Ablation	Validation F1		Test F1	
	Macro	Micro	Macro	Micro
Feature-based BERT with candidate sets	87.1±0.1	90.3±0.1	83.5±0.3	84.8 ±0.4
Feature-based BERT without candidate sets	63.3±1.1	64.1±0.2	57.2±0.2	54.1 ±0.3
With fasttext entity embedding	90.4	91.4	82.8	82.9

Table 2: Ablation results on validation and test sets of AIDA/CoNLL. By feature-based BERT we mean BERT which is not fine-tuned to the task.

Ablation Study. We perform a simple ablation study, the results of which are shown in Table 2. We note that performance suffers in the EL task when `BERT` is not fine-tuned but still maintains strong results comparable to the state-of-the-art. Without fine-tuning, validation set performance decreases and becomes more comparable to test set performance, indicating that the fine-tuned `BERT` overfits in such a setting - we find this to be an interesting future direction of study.

Other Results. Finally, during research, we swapped the Wikipedia2Vec entities with averaged-out 300-dimensional FastText embeddings (Bojanowski et al., 2017) to see what the impact of not having entity-specific embeddings would be. To our surprise, the model performs on par with existing results which, we think, points to a combination of (1) `BERT` already having internal knowledge of entity-mentions given their context; and (2) many AIDA mentions being easily linkable by simply considering their surface-form. We think this too is an interesting direction of future study. Point (2) specifically points to the need for better EL datasets than AIDA, which was originally meant to be an ED dataset. A great study of point (1) can be found in Poerner et al. (2019).

5 Conclusions and Future Work

We propose an EL model that jointly learns the MD and ED task, achieving state-of-the-art results. We also show that training and inference without candidate sets is possible. We think that interesting future directions of study include a better understanding of how BERT already comprehends entities in text without reference to external entity embeddings. Finally, we think that moving forward, reducing the EL community's dependence on candidate sets could be a good thing and requires more research. Dropping candidate sets could make models more easily comparable.

References

Ander Barrena, Aitor Soroa, and Eneko Agirre. 2018. Learning text representations for 500K classification tasks on named entity disambiguation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 171–180, Brussels, Belgium. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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

Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Chengjiang Li, Xu Chen, and Tiansi Dong. 2018. Joint representation learning of cross-lingual words and entities via attentive distant supervision. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 227–237, Brussels, Belgium. Association for Computational Linguistics.

Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. 2017. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1623–1633, Vancouver, Canada. Association for Computational Linguistics.

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

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. 2017. Named entity disambiguation for noisy text. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 58–68, Vancouver, Canada. 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, Copenhagen, Denmark. Association for Computational Linguistics.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Filip Ilievski, Piek Vossen, and Stefan Schlobach. 2018. Systematic study of long tail phenomena in entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 664–674, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

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

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia. Association for Computational Linguistics.

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

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

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

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics.

J. Redmon and A. Farhadi. 2017. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525.

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. GERBIL - benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2369–2374. ACM.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *16th International Conference on the World Wide Web*, pages 697–706.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2018. Wikipedia2vec: An optimized tool for learning embeddings of words and entities from wikipedia. *arXiv preprint 1812.06280*.

Ikuya Yamada and Hiroyuki Shindo. 2019. Pre-training of deep contextualized embeddings of words and entities for named entity disambiguation. *arXiv preprint arXiv:1909.00426*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany. Association for Computational Linguistics.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning distributed representations of texts and entities from knowledge base. *Transactions of the Association for Computational Linguistics*, 5:397–411.