Zero-shot Entity Linking with Efficient Long Range Sequence Modeling

Zonghai Yao
University of Massachusetts, Amherst
zonghaiyao@cs.umass.edu

Liangliang Cao
University of Massachusetts, Amherst
llcao@cs.umass.edu

Huapu Pan
Google Research
huapupan@google.com

Abstract

This paper considers the problem of zero-shot entity linking, in which a link in the test time may not present in training. Following the prevailing BERT-based research efforts, we find a simple yet effective way is to expand the long-range sequence modeling. Unlike many previous methods, our method does not require expensive pre-training of BERT with long position embeddings. Instead, we propose an efficient position embeddings initialization method called Embedding-repeat, which initializes larger position embeddings based on BERT-Base. On Wikia's zero-shot EL dataset, our method improves the SOTA from 76.06% to 79.08%, and for its long data, the corresponding improvement is from 74.57% to 82.14%. Our experiments suggest the effectiveness of long-range sequence modeling without retraining the BERT model. [1]

1 Introduction

Entity linking (EL) is the task of grounding entity mentions by linking them to entries in a given database or dictionary of entities. Traditional EL approaches often assume that entities linked at the test time are present in the training set. Nevertheless, many real-world applications prefer the zero-shot setting, where there is no external knowledge and a short text description provides the only information we have for each entity (Sil et al., 2012; Wang et al., 2015). For zero-shot entity linking (Logeswaran et al., 2019), it is crucial to consider the context of entity description and mention, so that the system can generalize to unseen entities. However, most of the BERT-based models are based on a context window with 512 tokens, limited to capturing the long-range of context. This paper defines a model's **Effective-Reading-Length (ER-Length)** as the total length of the mention contexts
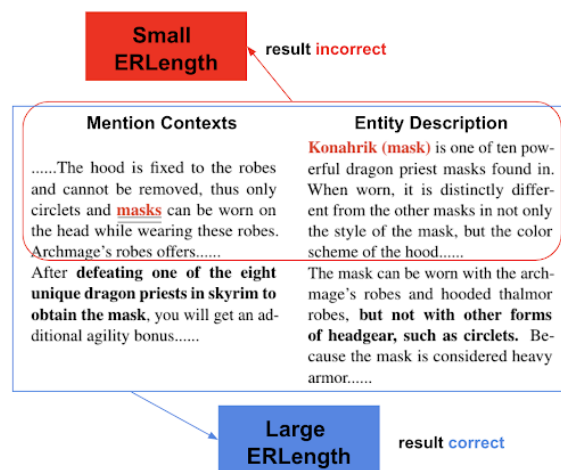

Figure 1: Only models with large ERLength can solve this entity linking problem because only they can get valuable critical information in the mention contexts and entity description.

and entity description that it can read. Figure 1 demonstrates an example where long ERlengths are more preferred than short ones.

Many existing methods that can be used to expand ERLength (Sohoni et al., 2019; Dai et al., 2019), however, often need to completely re-do pre-training with the masked language modeling objective on the vast general corpus (like Wikipedia), which is not only very expensive but also impossible in many scenarios.

This paper proposes a practical way, Embeddings-repeat, to expand BERT's ER-Length by initializing larger position embeddings, allowing reading all information in the context. Note our method differs from previous works since it can directly use the larger position embeddings initialized from BERT-Base to do fine-tuning on downstream tasks without any retraining. Extensive experiments are conducted to compare different ways of expanding ERLength, and the results show that Embeddings-repeat can robustly

---

[1] Our code are publicly available. https://github.com/seasonyao/Zero-Shot-Entity-Linking.

improve performance. Most importantly, we improve the accuracy from 76.06% to 79.08% in Wikia's zero-shot EL dataset, from 74.57% to 82.14% for its long data. Since our method is effective and easy to implement, we expect our method will be useful for other downstream NLP tasks.

2 Related work

Zero-shot Entity Linking Most state-of-the-art entity linking methods are composed of two steps: candidate generation (Sil et al., 2012; Vilnis et al., 2018; Radford et al., 2018) and candidate ranking (He et al., 2013; Sun et al., 2015; Yamada et al., 2016). Logeswaran et al. (2019) proposed the zero-shot entity linking task, where mentions must be linked to unseen entities without in-domain labeled data. For each mention, the model first uses BM25 (Robertson and Zaragoza, 2009) to generate 64 candidates. For each candidate, BERT (Devlin et al., 2018) will read a sequence pair combining mention contexts and entity description and produce a vector representation for it. Then, the model will rank the candidates based on these vectors. This paper discusses how to improve Logeswaran et al. (2019) by efficiently expanding the ERLength.

Modeling long documents The simplest way to work around the 512 limit is to truncate the document(Xie et al., 2019; Liu et al., 2019). It suffers from severe information loss, which does not meet sufficient information in the zero-shot entity linking. Recently there has been an explosive amount of efforts to improve long-range sequence modeling (Sukhbaatar et al., 2019; Rae et al., 2019; Child et al., 2019; Ye et al., 2019; Qiu et al., 2019; Lample et al., 2019; Lan et al., 2019). However, they all need to initialize new position embeddings and do expensive retraining on the general corpus (like Wikipedia) to learn the positional relationship in longer documents before fine-tuning downstream tasks. Moreover, the exploration of the impact of long-range sequence modeling on entity linking is still blank. So in this study, we will explore a different approach, which initializes larger position embeddings based on the existing small one in BERT-Base, and can be used directly in the fine-tuning without expensive retraining.
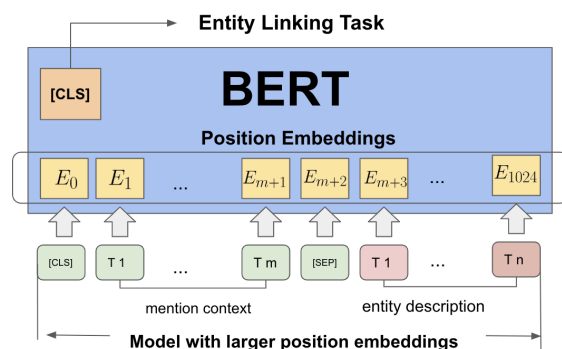

Figure 2: BERT doing entity linking with larger position embeddings

3 Method

3.1 Overview

Figure 2 describes how to use BERT for zero-shot entity linking tasks with larger position embeddings. Following Logeswaran et al. (2019), we adopt a two-stage pipeline consisting of a fast candidate generation stage, followed by a more expensive but powerful candidate ranking stage (Ganea and Hofmann, 2017; Kolitsas et al., 2018; Wu et al., 2019). We use BM25 for the candidate generation stage and get 64 candidate entities for every mention. For the candidate ranking stage, as in BERT, the mention contexts m and candidate entity description e are concatenated as a sequence pair together with special start and separator tokens: ([CLS] m [SEP] e [SEP]). The Transformer (Vaswani et al., 2017) will encode this sequence pair, and the position embeddings inside will capture the position information of individual words. At the last hidden layer, the Transformer produces a vector representation $h_{m,e}$ of the input pair through the special pooling token [CLS]. And then entities in a given candidate set are scored as $softmax(\omega^\top h_{m,e})$ where ω is a learned parameter vector.

Since the size of position embeddings is limited to 512 in BERT, how to capture position information beyond this size is what we hope to improve. In general, for new and larger position embeddings, we often need to re-initialize it with the larger size, and then retrain on general corpus like Wikipedia to learn the positional relationship in longer documents. However, we found that the relationship between different positions in the text is related. We can initialize larger position embeddings from the small ones in BERT-Base, and then without any expensive retraining, directly use it to complete the

Method	ERLength	eval						test
		set_1	set_2	set_3	set_4	avg	long	
Logeswaran et al. (2019)	256	83.40	79.00	73.03	68.82	76.06	74.57	75.06
BERT	512	83.45	80.03	71.88	72.53	76.97	78.54	-
Logeswaran et al. (2019) (DAP)	256	82.82	81.59	75.34	72.52	78.07	76.89	77.05
E_{repeat}	1024	87.02	81.52	73.48	74.37	79.08	82.14	77.58
E_{repeat} + DAP	1024	**89.67**	**83.53**	**75.37**	**74.96**	**80.88**	**82.14**	**79.64**

Table 1: Our methods with long ERlength outperform state of the art. Especially, the accuracy of the long data increases from 74.57% to 82.14% compared with the benchmark. Here, we call all data whose DLength exceeds 512 (the maximum number BERT can read) as **long** data. If we also use DAP, the best accuracy is 80.88% in validation data, and 79.64% in test data. Note: set_1: Coronation street, set_2: Muppets, set_3: Ice hockey, set_4: Elder scrolls, DAP: Domain Adaptive Pre-training (Logeswaran et al., 2019).

fine-tuning on the downstream tasks.

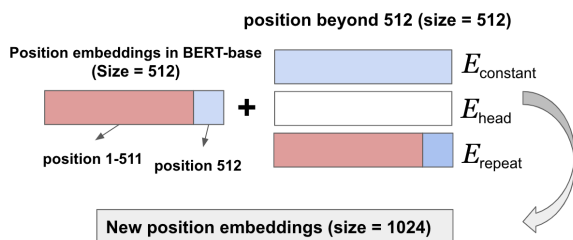### 3.2 Position embeddings initialization

Figure 3: BERT model with larger position embeddings which initialized from different method

It is reasonable to assume that the larger position embeddings have similar first 512 values with the small one since they all express the corresponding relationship between tokens when the input length is less than 512. For those positions over 512, we introduce a particular method **Embeddings-repeat** (E_{repeat}) to initialize larger position embeddings by repeating the small one from BERT-Base as analysis of BERT's attention heads shows a strong learned bias to attend to the local context, including the previous or next token (Clark et al., 2019). We assume using E_{repeat} preserves this local structure everywhere except at the partition boundaries. For example, for a 1024 position embeddings model, we will initialize the first 512 positions and the last 512 positions, respectively, from BERT-Base.

To verify the rationality of E_{repeat}, we also proposed two other methods as the comparison. E_{head} assumes only the first 512 positions in the larger position embeddings are similar to that in the small one, so it initializes the first 512 positions from BERT-Base and randomly initializes those exceeding 512. $E_{constant}$ also uses position embeddings in BERT-Base to initialize its first 512 positions.

However, it uses the value of position 512 to initialize those exceeding 512, since it assumes the relationship between two tokens over a long distance tend to be constant. In the following experimental part, we show that at least in this task, using E_{repeat} to expand the ERLength of BERT is most effective.

4 Experiments

4.1 Dataset and experiment setup

We use Wikia's zero-shot EL dataset constructed by Logeswaran et al. (2019), which to our knowledge, is the best zero-shot EL benchmark. To show the importance of long-range sequence modeling, we define the data's **DLength** as the total length of the mention contexts and entity description and examine the distribution of DLength on the dataset. As shown in Table 2, We found about half of the data have a DLength exceeding 512 tokens. Furthermore, 93% of them are less than 1024. So we set the model's ERLength range from 0 to 1024, with which we explore how continuously expanding the model's ERLength will affect its performance on Wikia's zero-shot EL dataset. When we increase ERLength, we will assign the same size growth to the mention contexts and entity description, which we find is the most reasonable through our related experiments.

For all experiments, we follow the most recent work in studying zero-shot entity linking. We use the BERT-Base model architecture in all our experiments. The Masked LM objective (Devlin et al., 2018) is used for unsupervised pre-training. For fine-tuning language models (in the case of multi-stage pre-training) and fine-tuning on the Entity-Linking task, we use a small learning rate

	Proportion of data in different DLength intervals									
DLength	(0,200)	[200,300)	[300,400)	[400,500)	[500,600)	[600,700)	[700,800)	[800,900)	[900,1000)	[1000,+∞)
% Of total	10.62	14.62	11.92	9.96	15.18	14.11	8.56	4.91	3.39	6.83
Accuracy for model with different ERLength on different DLength interval										
ERLength 64	60.76	62.10	62.20	58.61	64.18	61.50	62.45	60.52	65.41	62.57
128	73.57	70.57	71.31	67.64	72.34	69.34	70.87	70.12	68.49	71.69
256	75.52	74.16	75.50	73.34	75.35	75.47	75.11	73.71	77	74.34
384	75.72	77.11	78.44	74.03	78.00	75.84	77.07	78.44	75.51	78.96
512	76.64	75.81	78.00	74.68	77.99	77.33	78.97	81.34	75.48	79.49
768	75.56	75.66	79.08	75.15	77.87	78.54	78.54	82.11	78.22	79.96
1024	75.80	76.40	80.54	75.47	77.7	77.51	79.65	81.60	81.27	83.58

Table 2: The table shows the proportion of different DLength data and the accuracy of different ERLength models on different DLength data. Red represents the accuracy of the leading echelon in certain DLength data. It shows a cascading downward trend, which means that for larger DLength data, only models with larger ERLength can perform well, and even if ERLength is much larger than DLength, accuracy will not decline.

of 2e-5, following the recommendations from Devlin et al. (2018). All models are implemented in Tensorflow and optimized with Adam. All experiments were conducted with v3-8 TPU on Google Cloud.

Like Logeswaran et al. (2019), our entity linking performance is evaluated on the subset of test instances for which the gold entity is among the top-k candidates retrieved during candidate generation. Our IR-based candidate generation has a top-64 recall of 76% and 68% on the validation and test sets, respectively. Strengthening the candidate generation stage improves the final performance, but this is outside our work scope. Average performance across a set of domains is computed by macro-averaging. Performance is defined as the accuracy of the single-best identified entity (top-1 accuracy).

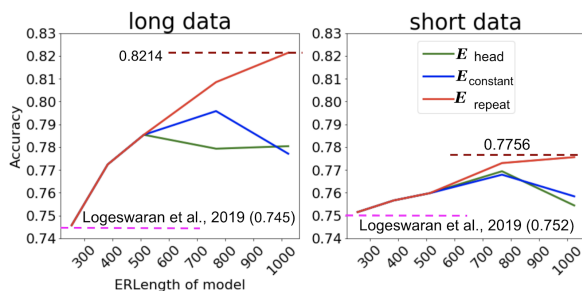### 4.2 Comparison of different initializations

Figure 4: The accuracy of the model with different position embeddings initialization methods in long and short data. Note: We call all data whose DLength exceeds 512 as **long** data, otherwise, **short** data.

The results of different position embeddings initialization methods are shown in figure 4. It can be found that for both long and short data, E_{repeat} has achieved the best results, especially its performance on long data is impressive. When the model's ERLength exceeds 512, only using E_{head} produces worse results, which shows the importance of using the information of the first 512 positions to initialize the latter part. The model with $E_{constant}$ starts to decrease after its ERLength reaches about 768, which shows that its assumption is only reasonable when the model's ERLength is less than 768. Only when using E_{repeat} to initialize we will see a stable and continuous improvement, which shows that only its "local structure" assumption applies to almost all theoretical lengths here (from 0 to about 1024). This also makes it an ideal method to explore the impact of increasing ERLength.

Table 1 suggests our method improves state of the art on Wikia's zero-shot EL dataset. Compared to Logeswaran et al. (2019), if we use E_{repeat} to increase the model's ERLength to 1024, we improve the accuracy from 76.06% to 79.08%, and for the long data, the improvement is from 74.57% to 82.14%. What's more, we also try the Domain Adaptive Pre-training (DAP) method in Logeswaran et al. (2019). The combination of DAP and 1024 ERLength raises the result to 80.88%.

4.3 Impact of increasing ERLlength

We further explore the impact of BERT's ERLength on the zero-shot EL task. The red in the table 2 represents the accuracies in the first echelon in each column (for data within a specific DLength interval). It shows a clear step-down trend, which means data with a larger DLength often requires a

model with a larger ERLength. What's more, for any column, if we continue to increase the model's ERLength, the accuracy will stabilize within a specific range after the ERLength exceeds most data's DLengths. So the last row in the table is always red, which means that the model with the largest ERLength can always achieve the best level of accuracy on all data of different DLengths.
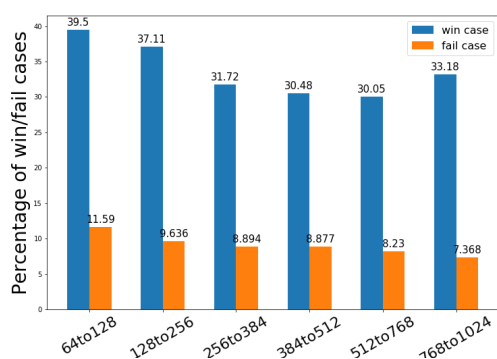

Figure 5: The proportion of win/fail cases during the increase in ERLength. We define **win** case as the initially wrong data but is now correct after increasing ERLength, and define **fail** case as the initially correct data is now wrong after increasing ERLength.

Figure 5 shows the changes of win and fail cases when expanding the BERT's ERLength. Generally speaking, when the model can read more content, its accuracy will increase for more valuable information (win case) and decrease for more noise (fail case). The results illustrate that BERT can always use more useful information to help itself while being less disturbed by noise. This once again demonstrates the power of the BERT's full-attention mechanism. This is also the basis on which we can continuously expand BERT's ERLength and continue to benefit. Therefore, for a particular dataset, when we set the ERLength of the BERT, letting it exceed more data's DLength can always bring more improvements.

Also, in the figure 6 we explore the importance of mention contexts and entity descriptions. On Wikia's zero-shot EL dataset, in our settings for BERT with 1024 ERLength, the mention contexts and entity description account for 512, respectively. In figure 6, if we unilaterally reduce the mention contexts and entity description from 512 to 50, the change of accuracy is shown in the figure. It can be found that the two are basically equally important, and no matter which side is reduced, the accuracy will gradually decrease. Therefore, when increas-
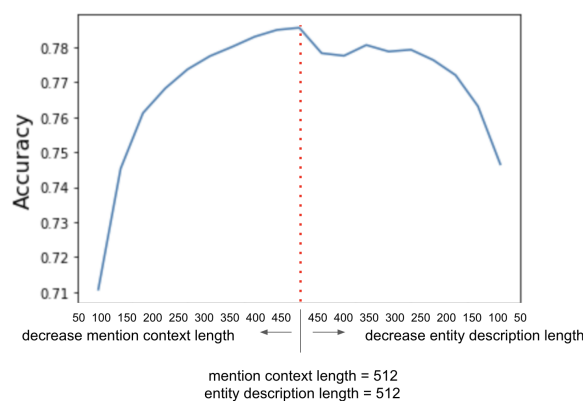

Figure 6: Importance of mention contexts and entity description

ing the BERT ERLength here, the best way is to increase the content of mention contexts and entity description at the same time.

5 Conclusions and future work

We propose an efficient position embeddings initialization method called Embeddings-repeat, which initializes larger position embeddings based on BERT models. For the zero-shot entity linking task, our method improves the SOTA from 76.06% to 79.08% on its dataset. Our experiments suggest the effectiveness of increasing ERLength as large as possible (e.g., the length of the longest data in the EL experiments). Our future work will be to extend our methods to other NLP tasks.

Acknowledgments

References

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

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

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

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

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

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large memory layers with product keys. In *Advances in Neural Information Processing Systems*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

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 Annual Meeting of the Association for Computational Linguistics*.

Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. 2019. Blockwise self-attention for long document understanding. *arXiv preprint arXiv:1911.02972*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI*.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

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*. Association for Computational Linguistics.

Nimit Sharad Sohoni, Christopher Richard Aberger, Megan Leszczynski, Jian Zhang, and Christopher Ré. 2019. Low-memory neural network training: A technical report. *arXiv preprint arXiv:1904.10631*.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Twenty-fourth international joint conference on artificial intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*.

Luke Vilnis, Irena Radovanovic, Andrew McCallum, et al. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. *arXiv preprint arXiv:1807.05127*.

Han Wang, Jin Guang 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*.

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

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.

Zihao Ye, Qipeng Guo, Quan Gan, Xipeng Qiu, and Zheng Zhang. 2019. Bp-transformer: Modelling long-range context via binary partitioning. *arXiv preprint arXiv:1911.04070*.