MixLinkBERT: A Language Model pretrained with Multiple Types of Linked Documents for Multi-hop Question Answering

Dongming Wu[1,2] Ryu Iida[1,2] Jong-Hoon Oh[2] Kentaro Torisawa[1,2]
[1]Nara Institute of Science and Technology, Graduate School of Science and Technology
[2] National Institute of Information and Communications Technology,
Data-driven Intelligent System Research Center
wu.dongming.wa9@is.naist.jp
{ryu.iida, rovellia, torisawa}@nict.go.jp

Abstract

Multi-hop question answering is the question answering that requires understanding and reasoning over multiple documents to find answers. In this work, we focus on improving language model pretraining to achieve better performance on downstream multi-hop question answering. We develop MixLinkBERT, a language model pretrained with multiple types of linked documents. We show that MixLinkBERT outperforms BERT and LinkBERT on HotpotQA.

1 Introduction

Multi-hop question answering (QA) is the question answering that requires finding, understanding and reasoning over multiple documents to find answers. Table 1 shows an example of multi-hop question in HotpotQA [1], a multi-hop QA dataset. To answer the question in this example, we need information of "the auto manufacturer headquarted in Minato, Tokyo, Japan" and "the auto manufacturer that Nissan acquired controlling interest in". These two pieces of information may exist in two different documents separately, such as the two Wikipedia articles shown in Table 1. In this case, finding, understanding and reasoning over these documents are necessary to answer this question.

The prevailing structure of QA systems for both conventional single-hop QA and multi-hop QA in these years mainly consists of two components, a retriever and a reader [2, 3, 4]. Given a question, a retriever first retrieves several candidate documents from a large set of documents, such as all Wikipedia articles, or documents collected from the internet. Then a reader extracts the answer from these candidate documents. Like many other NLP tasks, recent works of multi-hop QA [3, 4] fine-tuned pretrained language models (LMs), such as BERT [5], RoBERTa [6], ELECTRA [7], to use them as a retriever and a reader in their QA methods. For example, Xiong et al. [4] fine-tuned a pretrained RoBERTa as their retriever and fine-tuned a pretrained ELECTRA as their reader.

Table 1: An example of multi-hop question in HotpotQA [1].

Question: What Japanese auto manufacturer headquarted in Minato, Tokyo, Japan did Nissan acquire controlling interest in?
Document 1: [Mitsubishi Motors] Mitsubishi Motors Corporation is a Japanese multinational automotive manufacturer headquartered in Minato, Tokyo, Japan. In 2011, Mitsubishi Motors was ……
Document 2: [Renault–Nissan–Mitsubishi Alliance] The Renault-Nissan-Mitsubishi Alliance is a French-Japanese strategic partnership between automobile manufacturers …… The Alliance adopted its current name in September 2017, one year after Nissan acquired a controlling interest in Mitsubishi and ……
Answer: Mitsubishi.

To improve performance of multi-hop QA, new QA system structures or new fine-tuning methods have also been explored [3, 4]. In addition, pretraining LMs suitable for multi-hop reasoning on downstream tasks has been attempted. For example, Yasunaga et al. [8] developed LinkBERT, a LM that learned multi-hop knowledge (the

knowledge that spans across multiple documents) by leveraging hyperlinks in documents. The experimental results showed that LinkBERT achieved a better performance on multi-hop QA than BERT, which was pretrained without hyperlinks and cannot learn multi-hop knowledge.

In this work we develop a new language model pretrained with multiple types of linked documents, called MixLinkBERT. Along with the hyperlinks in the main texts in Wikipedia articles, which were used by LinkBERT, MixLinkBERT additionally uses hyperlinks in infoboxes in Wikipedia. An infobox in Wikipedia is a table often shown in upper right of a Wikipedia article and represents a summary of important information about the subject of an article by a set of attribute-value pairs. For example, Figure 1 shows part of the infobox of "Slam Dunk (manga)". The infobox contains attribute-value pairs like "genre-comedy", "author-Takehiko Inoue", "publisher-Shueisha". An important point here is that infoboxes sometimes contain useful hyperlinks that do not appear in the main texts of Wikipedia articles. Therefore, the text that is reachable via hyperlinks in infoboxes can introduce potentially useful knowledge that cannot be leveraged in LinkBERT.

We compare MixLinkBERT with BERT and LinkBERT on HotpotQA. MixLinkBERT outperforms BERT by +1.5% in answer F1 and +1.1% in joint F1, and outperforms LinkBERT by +0.8% in answer F1 and joint F1. These results suggest that a wider range of hyperlinks helps LMs learn a wider range of multi-hop knowledge.

Genre	Comedy[1] Coming-of-age[2] Sports[3]
Manga	
Written by	Takehiko Inoue
Published by	Shueisha
English publisher	AUS Madman Entertainment NA Viz Media Gutsoon! Entertainment (former) SG Chuang Yi
Imprint	Jump Comics
Magazine	*Weekly Shōnen Jump*

Figure 1: Part of the infobox of "Slam Dunk (manga)"

2 Related Work

2.1 LinkBERT

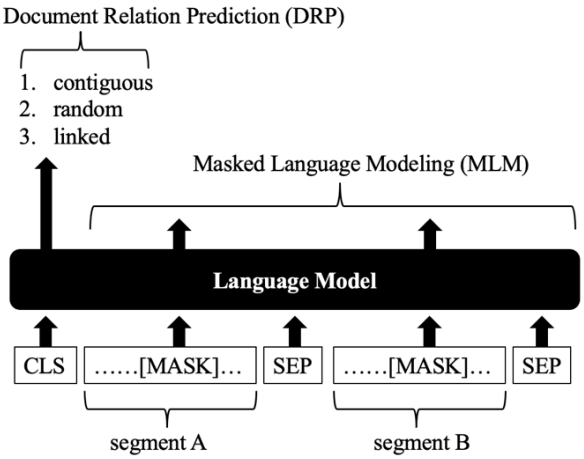

Figure 2: The overview of pretraining of LinkBERT [8]. (We drew this figure based on [8].)

Levine et al. [9] showed that in training of neural LMs, it is desirable to put a pair of text segments in the same training instance for learning strong dependencies between the pair of text. According to this observation, Yasunaga et al. [8] proposed LinkBERT, a LM that can learn multi-hop knowledge by leveraging hyperlinks in documents. Yasunaga et al. [8] placed text of hyperlinked documents in the same training instances, in addition to text of a single document or random documents as training instances of BERT [5]. Figure 2 shows the pretraining method of LinkBERT. LinkBERT has two training objectives, masked language modeling (MLM) and Document Relation Prediction (DRP). MLM objective is same as the MLM objective in BERT [5]. DRP classifies the relation of the two segments of an input sequence (segment A and segment B in Figure 2) into three classes: *contiguous*, *random* and *linked*. Corresponding to these three classes, there were three types of training instances in LinkBERT. In the following, we call these three types of training instances (1) contiguous-relation instances, (2) random-relation instances and (3) hyperlinked-from-text relation instances. Yasunaga et al. [8] created training instances for LinkBERT from Wikipedia as follows. To create each of their training instances, they first sampled an article from Wikipedia as document A, and a segment from document A as segment A. Then for each of (1) contiguous-relation instances, they sampled a contiguous

segment of segment A from the same document as segment B. For each of (2) random-relation instances, they randomly sampled another article from Wikipedia as document B and sampled a segment from document B as segment B. For each of (3) hyperlinked-from-text relation instances, they sampled an article that is hyperlinked from the main text of document A as document B and sampled a segment from document B as segment B. Then they concatenated segment A and segment B via special tokens to create a training instance: [CLS] Segment A [SEP] Segment B [SEP]. From Wikipedia, they created a set of training instances consists of 33% contiguous-relation instances, 33% random-relation instances and 33% hyperlinked-from-text relation instances.

2.2 Previous works using Wikipedia infoboxes

Information inside Wikipedia infoboxes is summarized by human and can be utilized for NLP tasks. Morales et al. [10] created INFOBOXQA, a question answering dataset, utilizing attribute and value information in infoboxes. Herzig et al. [11] used text in infoboxes directly as inputs for LM pretraining and improved the ability of QA over tables. Unlike Herzig et al. [11] using text in infoboxes directly as inputs for LM pretraining, we use text in articles that are hyperlinked via infoboxes as inputs for our LM pretraining since text in infoboxes usually only includes titles of articles or names of objects.

3 MixLinkBERT

3.1 Pretraining instances

To explore multi-hop knowledge that can be obtained from multiple types of linked documents, we utilize hyperlinks in Wikipedia infoboxes in addition to hyperlinks in the main texts of Wikipedia articles. More precisely, in addition to the three types of training instances used by LinkBERT, we add the fourth type of training instance: hyperlinked-from-infobox relation instance. To create each of hyperlinked-from-infobox relation instances, we first sample an article from Wikipedia as document A and a segment from document A as segment A. Then we sample an article that is hyperlinked from the infobox of document A as document B and sample a segment from document B as segment B. The proportion of each kind of training instances of MixLinkBERT is 33% for contiguous-relation instances, 33% for random-relation instances, 16.5% for hyperlinked-from-text relation instances, and 16.5% for hyperlinked-from-infobox relation instances.

3.2 Training objectives

We use the same training objectives of LinkBERT, masked language modeling (MLM) and Document Relation Prediction (DRP), to pretrain MixLinkBERT. For DRP objective, we set that both hyperlinked-from-text relation instances and hyperlinked-from-infobox relation instances belong to the same *linked* class.

4 Experiments

4.1 Setup for pretraining MixLinkBERT

We created pretraining data from English Wikipedia (20220820 version dump). Our pretraining data contains totally 40,960,000 training instances. We initialized the parameters of our LM with pretrained BERT-base-cased checkpoint released by Devlin et al. [5] and then started pretraining from the parameters.

4.2 Baselines

For a fair comparison between MixLinkBERT and other baseline models (BERT and LinkBERT), we also pretrained two baseline models by ourselves with same size of pretraining data of MixLinkBERT. For baseline BERT, we continued pretraining from pretrained BERT-base-cased checkpoint released by Devlin et al. [5] with original BERT's pretraining objectives. The pretraining data consists of 50% contiguous-relation instances and 50% random-relation instances. For baseline LinkBERT, we also started pretraining from pretrained BERT-base-cased checkpoint released by Devlin et al. [5] but with LinkBERT's [8] pretraining objectives. The only difference from pretraining of MixLinkBERT is that pretraining data of baseline LinkBERT doesn't contain hyperlinked-from-infobox relation instances. Pretraining data of baseline LinkBERT consists of 33% contiguous-relation instances, 33% random-relation instances, and 33% hyperlinked-from-text relation instances. Table 2 shows the difference of pretraining data of MixLinkBERT and the two baselines. We followed hyperparameter settings in Yasunaga et al. [8]. Training steps was 40,000, peak learning rate was 3e-4, batch size was 2,048, maximum sequence length was 512 tokens. We warmed up the learning rate for the first 5,000 steps and linearly

decayed it. Each pretraining took about 4 days on 8 32GB V100 GPUs with fp16.

Table 2: Pretraining data of MixLinkBERT and the baseline models

	training instances (percentage)			
	conti.	rand.	hyperlinked-from-text	hyperlinked-from-infobox
BERT (baseline)	50%	50%	0%	0%
LinkBERT (baseline)	33%	33%	33%	0%
MixLinkBERT	33%	33%	16.5%	16.5%

4.3 Evaluation on HotpotQA

We fine-tuned and evaluated MixLinkBERT and the two baselines on HotpotQA [1], a famous multi-hop QA dataset. Since ground-truth answers of test data of HotpotQA are not available, we instead split the development data of HotpotQA into two data sets and use one of the sets (3,702 instances) as our development data and another set (3,703 instances) as our test data. We fine-tuned with the method and the code[i] of a recent published multi-hop QA system [4]. We did hyperparameter search using our development data as follows. For retriever, we tried learning rate = {2e-5, 4e-5} and batch size = {75, 150, 300} and chose the best parameters that achieved the best Recall on top-2 retrieved documents on our development data. For reader, we tried learning rate = {3e-5, 5e-5, 1e-4} and batch size = {64, 128, 256} and chose the best parameters that achieved the best answer exact match on our development data. For rest of the fine-tuning hyperparameters, we followed the settings in Xiong et al. [4].

Table 3: Performance on our test data.

Pretrained LM for fine-tuning	Ans. EM	Ans. F1	Sup. EM	Sup. F1	Joint EM	Joint F1
BERT (baseline)	54.7	67.2	51.84	75.7	36.1	58.7
LinkBERT (baseline)	55.2	67.9	51.7	75.2	36.3	59.0
MixLinkBERT	**55.6**	**68.7**	**51.87**	**76.0**	**36.6**	**59.8**

[i]https://github.com/facebookresearch/multihop_dense_retrieval

4.4 Results

Table 3 shows the performance on our test data. MixLinkBERT outperforms baseline BERT notably, especially +1.5% on answer F1 and +1.1% on Joint F1. MixLinkBERT also outperforms baseline LinkBERT, with +0.8% on answer F1 and Joint F1.

As shown in the experimental results, pretraining with both hyperlinked-from-text relation instances and hyperlinked-from-infobox relation instances (MixLinkBERT) achieves better performance than pretraining without hyperlinked-from-infobox relation instances (baseline LinkBERT). This suggests that infoboxes contain useful hyperlinked articles that do not exist in the main texts of Wikipedia articles. In this work, we used 33%, 33%, 16.5%, 16.5% for contiguous-relation, random-relation, hyperlinked-from-text relation and hyperlinked-from-infobox relation instances respectively, but there is still room for exploring the optimal ratio of the training instance types. So, one of our next challenges is to find the best ratio of the four types of training instances to improve the performance on HotpotQA.

5 Conclusion

In this work, we focused on improving language model pretraining to achieve better performance on downstream multi-hop question answering. We developed MixLinkBERT, a language model pretrained with multiple types of linked documents to improve multi-hop reasoning ability on downstream multi-hop QA. We show that MixLinkBERT outperforms BERT and LinkBERT on HotpotQA. As future work, we also plan to incorporate attribute information in infoboxes into LM pretraining to further improve the performance on downstream multi-hop QA.

References

[1] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. **In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 2369–2380, 2018.

[2] Danqi Chen, Adam Fisch, Jason Weston, Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. **In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics**, pp. 1870–1879, 2017.

[3] Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, Caiming Xiong. Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering. **The International Conference on Learning Representations**, 2020.

[4] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, Barlas Oğuz. Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval. **The International Conference on Learning Representations**, 2021.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina 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**, pp. 4171–4186, 2019.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. **arXiv preprint arXiv:1907.11692**, 2019.

[7] Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. **The International Conference on Learning Representations**, 2020.

[8] Michihiro Yasunaga, Jure Leskovec, Percy Liang. LinkBERT: Pretraining Language Models with Document Links. **In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**, pp. 8003–8016, 2022.

[9] Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, Amnon Shashua. The Inductive Bias of In-Context Learning: Rethinking Pretraining Example Design. **The International Conference on Learning Representations**, 2022.

[10] Alvaro Morales, Varot Premtoon, Cordelia Avery, Sue Felshin, Boris Katz. Learning to Answer Questions from Wikipedia Infoboxes. **In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 1930–1935, 2016.

[11] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, Julian Martin Eisenschlos. TAPAS: Weakly Supervised Table Parsing via Pre-training. **In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4320–4333, 2020.