Abstract

Language model (LM) pretraining captures various knowledge from text corpora, helping downstream NLP tasks. However, existing methods such as BERT model a single document, failing to capture document dependencies and knowledge that spans across documents. In this work, we propose LinkBERT, an effective LM pretraining method that incorporates document links, such as hyperlinks. Given a pretraining corpus, we view it as a graph of documents, and create LM inputs by placing linked documents in the same context. We then train the LM with two joint self-supervised tasks: masked language modeling and our newly proposed task, document relation prediction. We study LinkBERT in two domains: general domain (pretrained on Wikipedia with hyperlinks) and biomedical domain (pretrained on PubMed with citation links). LinkBERT outperforms BERT on various downstream tasks in both domains. It is especially effective for multi-hop reasoning and few-shot QA (+5% absolute improvement on HotpotQA and TriviaQA), and our biomedical LinkBERT attains new state-of-the-art on various BioNLP tasks (+7% on BioASQ and USMLE). We release the pretrained models, LinkBERT and BioLinkBERT, as well as code and data.\(^1\)

1 Introduction

Pretrained language models (LMs), like BERT and GPTs (Devlin et al., 2019; Brown et al., 2020), have shown remarkable performance on many natural language processing (NLP) tasks, such as text classification and question answering (Raffel et al., 2020), becoming the foundation of modern NLP systems. By performing self-supervised learning on text, such as masked language modeling (Devlin et al., 2019), LMs learn to encode various knowledge from text corpora and produce informative language representations for downstream tasks (Bosselut et al., 2019; Bommasani et al., 2021).

However, existing LM pretraining methods typically consider a single document in each input context (Liu et al., 2019; Joshi et al., 2020), and do not model links between documents. This can pose limitations because documents often have rich dependencies with each other (e.g. hyperlinks, references), and knowledge can span across documents. As a simple example, in Figure 1, the Wikipedia article “Tidal Basin, Washington D.C.” (left) describes that the basin hosts “National Cherry Blossom Festival”. The hyperlinked article (right) reveals the background that the festival celebrates “Japanese cherry trees”. Taken together, the link offers new knowledge not available in a single document (e.g. “Tidal Basin has Japanese cherry trees”), which can be useful for various applications, including answering a question “What trees can you see at Tidal Basin?”. We aim to leverage document links to incorporate more knowledge into language model pretraining.

Figure 1: Document links (e.g. hyperlinks) can provide salient multi-hop knowledge. For instance, the Wikipedia article “Tidal Basin” (left) describes that the basin hosts “National Cherry Blossom Festival”. The hyperlinked article (right) reveals the background that the festival celebrates “Japanese cherry trees”. Taken together, the link offers new knowledge not available in a single document (e.g. “Tidal Basin has Japanese cherry trees”), which can be useful for various applications, including answering a question “What trees can you see at Tidal Basin?”. We aim to leverage document links to incorporate more knowledge into language model pretraining.

\(^1\)Available at https://github.com/michiyasunaga/LinkBERT.

---

LinkBERT: Pretraining Language Models with Document Links

Michihiro Yasunaga  Jure Leskovec*  Percy Liang*
Stanford University
{myasu,jure,pliang}@cs.stanford.edu
even make discoveries too (Margolis et al., 1999).

In this work, we propose LinkBERT, an effective language model pretraining method that incorporates document link knowledge. Given a text corpus, we obtain links between documents such as hyperlinks, and create LM inputs by placing linked documents in the same context window, besides the existing option of placing a single document or random documents as in BERT. Specifically, as in Figure 2, after sampling an anchor text segment, we place either (1) the contiguous segment from the same document, (2) a random document, or (3) a document linked from anchor segment, as the next segment in the input. We then train the LM with two joint objectives: We use masked language modeling (MLM) to encourage learning multi-hop knowledge of concepts brought into the same context by document links (e.g. “Tidal Basin” and “Japanese cherry” in Figure 1). Simultaneously, we propose a Document Relation Prediction (DRP) objective, which classifies the relation of the second segment to the first segment (contiguous, random, or linked). DRP encourages learning the relevance and bridging concepts (e.g. “National Cherry Blossom Festival”) between documents, beyond the ability learned in the vanilla next sentence prediction objective in BERT.

Viewing the pretraining corpus as a graph of documents, LinkBERT is also motivated as self-supervised learning on the graph, where DRP and MLM correspond to link prediction and node feature prediction in graph machine learning (Yang et al., 2015; Hu et al., 2020). Our modeling approach thus provides a natural fusion of language-based and graph-based self-supervised learning.

We train LinkBERT on two domains: the general domain, using Wikipedia articles with hyperlinks (§4), and the biomedical domain, using PubMed articles with citation links (§6). We then evaluate the pretrained models on a wide range of downstream tasks including question answering, in both domains. LinkBERT consistently improves on baseline LMs across domains and tasks. For the general domain, LinkBERT outperforms BERT on MRQA benchmark (+4% absolute in F1-score) as well as GLUE benchmark. For the biomedical domain, LinkBERT exceeds PubmedBERT (Gu et al., 2020) and attains new state-of-the-art on BLURB biomedical NLP benchmark (+3% absolute in BLURB score) and MedQA-USMLE reasoning task (+7% absolute in accuracy). Overall, LinkBERT attains notably large gains for multi-hop reasoning, multi-document understanding, and few-shot question answering, suggesting that LinkBERT internalizes significantly more knowledge than existing LMs by pretraining with document link information.

2 Related work

Retrieval-augmented LMs. Several works (Lewis et al., 2020b; Karpukhin et al., 2020; Oguz et al., 2020; Xie et al., 2022) introduce a retrieval module for LMs, where given an anchor text (e.g. question), retrieved text is added to the same LM context to improve model inference (e.g. answer prediction). These works show the promise of placing related documents in the same LM context at inference time, but they do not study pretraining. Guu et al. (2020) pretrain an LM with a retriever that learns to retrieve text for answering masked tokens in the anchor text. In contrast, our focus is not on retrieval, but on pretraining a general-purpose LM that internalizes knowledge that spans across documents, which is orthogonal to the above works (e.g., our pretrained LM could be used to initialize the LM component of these works). Additionally, we focus on incorporating document links such as hyperlinks, which can offer salient knowledge that common lexical retrieval methods may not provide (Asai et al., 2020).

Pretrain LMs with related documents. Several concurrent works use multiple related documents
to pretrain LMs. Caciularu et al. (2021) place documents (news articles) about the same topic into the same LM context, and Levine et al. (2021) place sentences of high lexical similarity into the same context. Our work provides a general method to incorporate document links into LM pretraining, where lexical or topical similarity can be one instance of document links, besides hyperlinks. We focus on hyperlinks in this work, because we find they can bring in salient knowledge that may not be obvious via lexical similarity, and yield a more performant LM (§5.5). Additionally, we propose the DRP objective, which improves modeling multiple documents and relations between them in LMs (§5.5).

Hyperlinks and citation links for NLP. Hyperlinks are often used to learn better retrieval models. Chang et al. (2020); Asai et al. (2020); Seonwoo et al. (2021) use Wikipedia hyperlinks to train retrievers for open-domain question answering. Ma et al. (2021) study various hyperlink-aware pretrained tasks for retrieval. While these works use hyperlinks to learn retrievers, we focus on using hyperlinks to create better context for learning general-purpose LMs. Separately, Calixto et al. (2021) use Wikipedia hyperlinks to learn multilingual LMs. Citation links are often used to improve summarization and recommendation of academic papers (Qazvinian and Radev, 2008; Yasunaga et al., 2019; Bhagavatula et al., 2018; Khadka et al., 2020; Cohan et al., 2020). Here we leverage citation link information into LM pretraining. Our approach (§4) will build on MLM and NSP.

4 LinkBERT

We present LinkBERT, a self-supervised pretraining approach that aims to internalize more knowledge into LMs using document link information. Specifically, as shown in Figure 2, instead of viewing the pretraining corpus as a set of documents $X = \{X^{(i)}\}$, we view it as a graph of documents, $G = (X, E)$, where $E = \{(X^{(i)}, X^{(j)})\}$ denotes links between documents (§4.1). The links can be existing hyperlinks, or could be built by other methods that capture document relevance. We then consider pretraining tasks for learning from document links (§4.2): We create LM inputs by placing linked documents in the same context window, besides the existing options of a single document or random documents. We use the MLM task to learn concepts brought together in the context by document links, and we also introduce the Document Relation Prediction (DRP) task to learn relations between documents. Finally, we discuss strategies for obtaining informative pairs of linked documents to feed into LM pretraining (§4.3).

4.1 Document graph

Given a pretraining corpus, we link related documents so that the links can bring together knowledge that is not available in single documents. We focus

\[ f_{\text{head}}(f_{\text{enc}}(X)) \]

\[ X = (x_1, x_2, \ldots, x_n) \]

\[ \{h_1, h_2, \ldots, h_n\} \]

A segment is typically a sentence or a paragraph.

\[ \text{NSP} \]

\[ \text{MLM} \]

\[ \text{MLM} \]

\[ \text{DRP} \]

\[ \text{DRP} \]
on hyperlinks, e.g., hyperlinks of Wikipedia articles (§5) and citation links of academic articles (§6). Hyperlinks have a number of advantages. They provide background knowledge about concepts that the document writers deemed useful—the links are likely to have high precision of relevance, and can also bring in relevant documents that may not be obvious via lexical similarity alone (e.g., in Figure 1, while the hyperlinked article mentions “Japanese” and “Yoshino” cherry trees, these words do not appear in the anchor article). Hyperlinks are also ubiquitous on the web and easily gathered at scale (Aghajanyan et al., 2021). To construct the document graph, we simply make a directed edge \((X^{(i)}, X^{(j)})\) if there is a hyperlink from document \(X^{(i)}\) to document \(X^{(j)}\).

For comparison, we also experiment with a document graph built by lexical similarity between documents. For each document \(X^{(i)}\), we use the common TF-IDF cosine similarity metric (Chen et al., 2017; Yasunaga et al., 2017) to obtain top-\(k\) documents \(X^{(j)}\)’s and make edges \((X^{(i)}, X^{(j)})\). We use \(k = 5\).

4.2 Pretraining tasks

Creating input instances. Several works (Gao et al., 2021; Levine et al., 2021) find that LMs can learn stronger dependencies between words that were shown together in the same context during training, than words that were not. To effectively learn knowledge that spans across documents, we create LM inputs by placing linked documents in the same context window, besides the existing option of a single document or random documents. Specifically, we first sample an anchor text segment from the corpus (Segment A; \(X_A \subseteq X^{(i)}\)). For the next segment (Segment B; \(X_B\)), we either (1) use the contiguous segment from the same document \(X_B \subseteq X^{(i)}\), (2) sample a segment from a random document \(X_B \subseteq X^{(j)}\) where \(j \neq i\), or (3) sample a segment from one of the documents linked from Segment A \(X_B \subseteq X^{(j)}\) where \((X^{(i)}, X^{(j)}) \in E\). We then join the two segments via special tokens to form an input instance: \([CLS] X_A [SEP] X_B [SEP]\).

Training objectives. To train the LM, we use two objectives. We apply the MLM objective to encourage the LM to learn multi-hop knowledge of concepts brought together in the same context by document links. We also propose a Document Relation Prediction (DRP) objective, which classifies the relation \(r\) of segment \(X_B\) to segment \(X_A\) \((r \in \{\text{contiguous, random, linked}\})\). By distinguishing \(\text{linked}\) from \(\text{contiguous}\) and \(\text{random}\), DRP encourages the LM to learn the relevance and existence of bridging concepts between documents, besides the capability learned in the vanilla NSP objective.

To predict \(r\), we use the representation of \([CLS]\) token, as in NSP. Taken together, we optimize:

\[
\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{DRP}}
\]

\[
= - \sum_i \log p(x_i | h_i) - \log p(r | h_{[CLS]})
\]

where \(x_i\) is each token of the input instance, \([CLS]\) \(X_A [SEP] X_B [SEP]\), and \(h_i\) is its representation.

4.3 Strategy to obtain linked documents

As described in §4.1, §4.2, our method builds links between documents, and for each anchor segment, samples a linked document to put together in the LM input. Here we discuss three key axes to consider to obtain useful linked documents in this process.

Relevance. Semantic relevance is a requisite when building links between documents. If links were randomly built without relevance, LinkBERT would be same as BERT, with simply two options of LM inputs (\text{contiguous or random}). Relevance can be achieved by using hyperlinks or lexical similarity metrics, and both methods yield substantially better performance than using random links (§5.5).

Salience. Besides relevance, another factor to consider (salience) is whether the linked document can offer new, useful knowledge that may not be obvious to the current LM. Hyperlinks are potentially more advantageous than lexical similarity links in this regard: LMs are shown to be good at recognizing lexical similarity (Zhang et al., 2020), and hyperlinks can bring in useful background knowledge that...
may not be obvious via lexical similarity alone (Asai et al., 2020). Indeed, we empirically find that using hyperlinks yields a more performant LM (§5.5).

**Diversity.** In the document graph, some documents may have a very high in-degree (e.g., many incoming hyperlinks, like the “United States” page of Wikipedia), and others a low in-degree. If we uniformly sample from the linked documents for each anchor segment, we may include documents of high in-degree too often in the overall training data, losing diversity. To adjust so that all documents appear with a similar frequency in training, we sample a linked document with probability inversely proportional to its in-degree, as done in graph data mining literature (Henzinger et al., 2000). We find that this technique yields a better LM performance (§5.5).

5 Experiments

We experiment with our proposed approach in the general domain first, where we pretrain LinkBERT on Wikipedia articles with hyperlinks (§5.1) and evaluate on a suite of downstream tasks (§5.2). We compare with BERT (Devlin et al., 2019) as our baseline. We experiment in the biomedical domain in §6.

5.1 Pretraining setup

**Data.** We use the same pretraining corpus used by BERT: Wikipedia and BookCorpus (Zhu et al., 2015). For Wikipedia, we use the WikiExtractor³ to extract hyperlinks between Wiki articles. We then create training instances by sampling contiguous, random, or linked segments as described in §4, with the three options appearing uniformly (33%, 33%, 33%). For BookCorpus, we create training instance by sampling contiguous or random segments (50%, 50%) as in BERT. We then combine the training instances from Wikipedia and BookCorpus to train LinkBERT. In summary, our pretraining data is the same as BERT, except that we have hyperlinks between Wikipedia articles.

**Implementation.** We pretrain LinkBERT of three sizes, -tiny, -base and -large, following the configurations of BERT\_tiny (4.4M parameters), BERT\_base (110M params), and BERT\_large (340M params) (Devlin et al., 2019; Turc et al., 2019). We use -tiny mainly for ablation studies.

For -tiny, we pretrain from scratch with random weight initialization. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with \((\beta_1, \beta_2) = (0.9, 0.98)\), warm up the learning rate for the first 5,000 steps and then linearly decay it.

We train for 10,000 steps with a peak learning rate 5e-3, weight decay 0.01, and batch size of 2,048 sequences with 512 tokens. Training takes 1 day on two GeForce RTX 2080 Ti GPUs with fp16.

For -base, we initialize LinkBERT with the BERT\_base checkpoint released by Devlin et al. (2019) and continue pretraining. We use a peak learning rate 3e-4 and train for 40,000 steps. Other training hyperparameters are the same as -tiny. Training takes 4 days on four A100 GPUs with fp16.

For -large, we follow the same procedure as -base, except that we use a peak learning rate of 2e-4. Training takes 7 days on eight A100 GPUs with fp16.

**Baselines.** We compare LinkBERT with BERT. Specifically, for the -tiny scale, we compare with BERT\_tiny, which we pretrain from scratch with the same hyperparameters as LinkBERT\_tiny. The only difference is that LinkBERT uses document links to create LM inputs, while BERT does not.

For -base scale, we compare with BERT\_base, for which we take the BERT\_base release by Devlin et al. (2019) and continue pretraining it with the vanilla BERT objectives on the same corpus for the same number of steps as LinkBERT\_base.

For -large, we follow the same procedure as -base.

5.2 Evaluation tasks

We fine-tune and evaluate LinkBERT on a suite of downstream tasks.

**Extractive question answering (QA).** Given a document (or set of documents) and a question as input, the task is to identify an answer span from the document. We evaluate on six popular datasets from the MRQA shared task (Fisch et al., 2019): HotpotQA (Yang et al., 2018), TriviaQA (Joshi et al., 2017), NaturalQ (Kwiatkowski et al., 2019), SearchQA (Dunn et al., 2017), NewsQA (Trischler et al., 2017), and SQuAD (Rajpurkar et al., 2016). As the MRQA shared task does not have a public test set, we split the dev set in half to make new dev and test sets. We follow the fine-tuning method BERT (Devlin et al., 2019) and continue pretraining. We use -tiny mainly for ablation studies.

For -tiny, we pretrain from scratch with random weight initialization. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with \((\beta_1, \beta_2) = (0.9, 0.98)\), warm up the learning rate for the first 5,000 steps and then linearly decay it.

We train for 10,000 steps with a peak learning rate 5e-3, weight decay 0.01, and batch size of 2,048 sequences with 512 tokens. Training takes 1 day on two GeForce RTX 2080 Ti GPUs with fp16.

For -base, we initialize LinkBERT with the BERT\_base checkpoint released by Devlin et al. (2019) and continue pretraining. We use a peak learning rate 3e-4 and train for 40,000 steps. Other training hyperparameters are the same as -tiny. Training takes 4 days on four A100 GPUs with fp16.

For -large, we follow the same procedure as -base, except that we use a peak learning rate of 2e-4. Training takes 7 days on eight A100 GPUs with fp16.

**Baselines.** We compare LinkBERT with BERT. Specifically, for the -tiny scale, we compare with BERT\_tiny, which we pretrain from scratch with the same hyperparameters as LinkBERT\_tiny. The only difference is that LinkBERT uses document links to create LM inputs, while BERT does not.

For -base scale, we compare with BERT\_base, for which we take the BERT\_base release by Devlin et al. (2019) and continue pretraining it with the vanilla BERT objectives on the same corpus for the same number of steps as LinkBERT\_base.

For -large, we follow the same procedure as -base.

**GLUE.** The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) is a popular suite of sentence-level classification tasks. Following BERT, we evaluate on CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), QQP, STS-B (Cer et al., 2017), MNLI (Williams et al., 2017), QNLI (Rajpurkar et al., 2016), and RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo
Table 1: Performance (F1) on MRQA question answering datasets. LinkBERT consistently outperforms BERT on all datasets across the -tiny, -base, and -large scales. The gain is especially large on datasets that require reasoning with multiple documents in the context, such as HotpotQA, TriviaQA, SearchQA.

Table 2: Performance on the GLUE benchmark. LinkBERT attains comparable or moderately improved performance.

Table 3: Performance (F1) on SQuAD when distracting documents are added to the context. While BERT incurs a large drop in F1, LinkBERT does not, suggesting its robustness in understanding document relations.

Table 4: Few-shot QA performance (F1) when 10% of fine-tuning data is used. LinkBERT attains large gains, suggesting that it internalizes more knowledge than BERT in pretraining.

Table 5: Ablation study on what linked documents to feed into LM pretraining (§4.3).

Table 6: Ablation study on the document relation prediction (DRP) objective in LM pretraining (§4.2).

Improved multi-hop reasoning. In Table 1, we find that LinkBERT obtains notably large gains on QA datasets that require reasoning with multiple documents, such as HotpotQA (+5% over BERT$_{tiny}$), TriviaQA (+6%) and SearchQA (+8%), as opposed to SQuAD (+1.4%) which just has a single document per question. To further gain qualitative insights, we studied in what QA examples LinkBERT succeeds but BERT fails. Figure 3 shows a representative example from HotpotQA.

Improved understanding of document relations. While the MRQA datasets typically use ground-truth documents as context for answering questions, in open-domain QA, QA systems need to use documents obtained by a retriever, which may include noisy documents besides gold ones (Chen et al., 2017; Dunn et al., 2017). In such cases, QA systems need to understand the document relations to perform well (Yang et al., 2018). To simulate this setting, we modify the SQuAD dataset such that we prepend or append 1–2 distracting documents to the original document given to each question. Table 3 shows the result. While BERT incurs a large performance drop (~2.8%), LinkBERT is robust to distracting documents (~0.5%). This result suggests that pretraining with document links improves the ability to understand document relations and

et al., 2007), and report the average score. More fine-tuning details are provided in Appendix B.

5.3 Results
Table 1 shows the performance (F1 score) on MRQA datasets. LinkBERT substantially outperforms BERT on all datasets. On average, the gain is +4.1% absolute for the BERT$_{tiny}$ scale, +2.6% for the BERT$_{base}$ scale, and +2.5% for the BERT$_{large}$ scale. Table 2 shows the results on GLUE, where LinkBERT performs moderately better than BERT. These results suggest that LinkBERT is especially effective at learning knowledge useful for QA tasks (e.g., world knowledge), while keeping performance on sentence-level language understanding too.

5.4 Analysis
We further study when LinkBERT is especially useful in downstream tasks.
relevance. In particular, our intuition is that the DRP objective helps the LM to better recognize document relations like (anchor document, linked document) in pretraining, which helps to recognize relations like (question, right document) in downstream QA tasks. We indeed find that ablating the DRP objective from LinkBERT hurts performance (§5.5). The strength of understanding document relations also suggests the promise of applying LinkBERT to various retrieval-augmented methods and tasks (e.g. Lewis et al. 2020b), either as the main LM or the dense retriever component.

Improved few-shot QA performance. We also find that LinkBERT is notably good at few-shot learning. Concretely, for each MRQA dataset, we fine-tune with only 10% of the available training data, and report the performance in Table 4. In this few-shot regime, LinkBERT attains more significant gains over BERT, compared to the full-resource regime in Table 1 (on NaturalQ, 5.4% vs 1.8% absolute in F1, or 15% vs 7% in relative error reduction). This result suggests that LinkBERT internalizes more knowledge than BERT during pretraining, which supports our core idea that document links can bring in new, useful knowledge for LMs.

5.5 Ablation studies
We conduct ablation studies on the key design choices of LinkBERT.

What linked documents to feed into LMs? We study the strategies discussed in §4.3 for obtaining linked documents: relevance, salience, and diversity. Table 5 shows the ablation result on MRQA datasets. First, if we ignore relevance and use random document links instead of hyperlinks, we get the same performance as BERT (−4.1% on average; “random” in Table 5). Second, using lexical similarity links instead of hyperlinks leads to 1.8% performance drop (“TF-IDF”). Our intuition is that hyperlinks can provide more salient knowledge that may not be obvious via lexical similarity alone. Nevertheless, using lexical similarity links is substantially better than BERT (+2.3%), confirming the efficacy of placing relevant documents together in the input for LM pretraining. Finally, removing the diversity adjustment in document sampling leads to 1% performance drop (“No diversity”). In summary, our insight is that to create informative inputs for LM pretraining, the linked documents must be semantically relevant and ideally be salient and diverse.

Effect of the DRP objective. Table 6 shows the ablation result on the DRP objective (§4.2). Removing DRP in pretraining hurts downstream QA performance. The drop is large on tasks with multiple documents (HotpotQA, TriviaQA, and SQuAD with distracting documents). This suggests that DRP facilitates LMs to learn document relations.

6 Biomedical LinkBERT (BioLinkBERT)
Pretraining LMs on biomedical text is shown to boost performance on biomedical NLP tasks (Beltagy et al., 2019; Lee et al., 2020; Lewis et al., 2020a; Gu et al., 2020). Biomedical LMs are typically trained on PubMed, which contains abstracts and citations of biomedical papers. While prior works only use their raw text for pretraining, academic papers have rich dependencies with each other via citations (references). We hypothesize that incorporating citation links can help LMs learn dependencies between papers and knowledge that spans across them.

With this motivation, we pretrain LinkBERT on PubMed with citation links (§6.1), which we term BioLinkBERT, and evaluate on biomedical downstream tasks (§6.2). As our baseline, we follow and compare with the state-of-the-art biomedical LM, PubmedBERT (Gu et al., 2020), which has the same architecture as BERT and is trained on PubMed.

6.1 Pretraining setup
Data. We use the same pretraining corpus used by PubmedBERT: PubMed abstracts (21GB).4 We

4https://pubmed.ncbi.nlm.nih.gov. We use papers published before Feb. 2020 as in PubmedBERT.
use the Pubmed Parser\textsuperscript{5} to extract citation links between articles. We then create training instances by sampling contiguous, random, or linked segments as described in §4, with the three options appearing uniformly (33\%, 33\%, 33\%). In summary, our pre-training data is the same as PubmedBERT, except that we have citation links between PubMed articles.

**Implementation.** We pretrain BioLinkBERT of -base size (110M params) from scratch, following the same hyperparameters as the PubmedBERT\textsubscript{base} (Gu et al., 2020). Specifically, we use a peak learning rate 6e-4, batch size 8,192, and train for 62,500 steps. We warm up the learning rate in the first 10% of steps and then linearly decay it. Training takes 7 days on eight A100 GPUs with fp16.

Additionally, while the original PubmedBERT release did not include the -large size, we pretrain BioLinkBERT of the -large size (340M params) from scratch, following the same procedure as -base, except that we use a peak learning rate of 4e-4 and warm up steps of 20%. Training takes 21 days on eight A100 GPUs with fp16.

**Baselines.** We compare BioLinkBERT with PubmedBERT released by Gu et al. (2020).

### 6.2 Evaluation tasks

For downstream tasks, we evaluate on the BLURB benchmark (Gu et al., 2020), a diverse set of biomedical NLP datasets, and MedQA-USMLE (Jin et al., 2021), a challenging biomedical QA dataset.

**BLURB** consists of five named entity recognition tasks, a PICO (population, intervention, comparison, and outcome) extraction task, three relation extraction tasks, a sentence similarity task, a document classification task, and two question answering tasks, as summarized in Table 7. We follow the same fine-tuning method and evaluation metric used by PubmedBERT (Gu et al., 2020).

**MedQA-USMLE** is a 4-way multi-choice QA task that tests biomedical and clinical knowledge. The questions are from practice tests for the US Medical License Exams (USMLE). The questions typically require multi-hop reasoning, e.g., given patient symptoms, infer the likely cause, and then answer the appropriate diagnosis procedure (Figure 4). We follow the fine-tuning method in Jin et al. (2021). More details are provided in Appendix B.

**MMLU-professional medicine** is a multi-choice QA task that tests biomedical knowledge and reasoning, and is part of the popular MMLU benchmark (Hendrycks et al., 2021) that is used to evaluate massive language models. We take the BioLinkBERT fine-tuned on the above MedQA-USMLE task, and evaluate on this task without further adaptation.

### 6.3 Results

**BLURB.** Table 7 shows the results on BLURB. BioLinkBERT\textsubscript{base} outperforms PubmedBERT\textsubscript{base} on all task categories, attaining a performance boost of +2% absolute on average. Moreover, BioLinkBERT\textsubscript{large} provides a further boost of +1%. In total, BioLinkBERT outperforms the previous best by +3% absolute, establishing new state-of-the-art on the BLURB leaderboard. We see a trend that gains are notably large on document-level tasks such as question answering (+7% on BioASQ and

---

\textsuperscript{5}https://github.com/titipata/pubmed_parser

---

### Table 7: Performance on BLURB benchmark. BioLinkBERT attains improvement on all tasks, establishing new state-of-the-art on BLURB. Gains are notably large on document-level tasks such as PubMedQA and BioASQ.

| Methods | Acc. (%) |
|---------|----------|
| BioBERT\textsubscript{large} (Lee et al., 2020) | 36.7 |
| QAGNN (Yasunaga et al., 2021) | 38.0 |
| GreaseLM (Zhang et al., 2022) | 38.5 |
| PubmedBERT\textsubscript{base} (Gu et al., 2020) | 38.1 |
| BioLinkBERT\textsubscript{base} (Ours) | 40.0 |
| BioLinkBERT\textsubscript{large} (Ours) | 44.6 |

---

### Table 8: Performance on MedQA-USMLE. BioLinkBERT outperforms all previous biomedical LMs.

| Methods | Acc. (%) |
|---------|----------|
| GPT-3 (175B params) (Brown et al., 2020) | 38.7 |
| UnifiedQA (11B params) (Khashabi et al., 2020) | 43.2 |
| BioLinkBERT\textsubscript{large} (Ours) | 50.7 |

---

Table 7: Performance on BLURB benchmark. BioLinkBERT attains improvement on all tasks, establishing new state-of-the-art on the BLURB leaderboard. Gains are notably large on document-level tasks such as PubMedQA and BioASQ.
Three days after undergoing a laparoscopic Whipple’s procedure, a 43-year-old woman has swelling of her right leg. ... … She was diagnosed with pancreatic cancer 1 month ago. ... Her temperature is 38°C (100.4°F), pulse is 90/min, and blood pressure is 118/78 mm Hg. Examination shows mild swelling of the right thigh to the ankle; there is no edema or pitting edema. ... Which of the following is the most appropriate next step in management?

(A) CT pulmonary angiography  (B) Compression ultrasonography  
(C) D-dimer level  (D) 2 sets of blood cultures

Figure 4: Case study of multi-hop reasoning on MedQA-USMLE. Answering the question (left) needs 2-hop reasoning (center): from the patient symptoms described in the question (leg swelling, pancreatic cancer), infer the cause (deep vein thrombosis), and then infer the appropriate diagnosis procedure (compression ultrasonography). While the existing PubmedBERT tends to simply predict a choice that contains a word appearing in the question (“blood” for choice D), BioLinkBERT correctly predicts the answer (B). Our intuition is that citation links bring relevant documents together in the same context in pretraining (right), which readily provides the multi-hop knowledge needed for the reasoning (center).

For instance, as in Figure 4 (right), Ansari et al. (2015) in PubMed mention that "pancreatic cancer can induce deep vein thrombosis in leg," and it cites a paper in PubMed, Piovella et al. (2002), which mentions that "deep vein thrombosis is tested by compression ultrasonography." Placing these two documents in the same context yields the complete multi-hop knowledge needed to answer the question (“pancreatic cancer” → “deep vein thrombosis” → “compression ultrasonography”). BioLinkBERT<sub>large</sub> achieves 50% accuracy on this QA task, significantly outperforming the largest general-domain LM or QA models such as GPT-3 175B params (39% accuracy) and UnifiedQA 11B params (43% accuracy). This result shows that with an effective pretraining approach, a small domain-specialized LM can outperform orders of magnitude larger language models on QA tasks.

7 Conclusion

We presented LinkBERT, a new language model (LM) pretraining method that incorporates document link knowledge such as hyperlinks. In both the general domain ( pretrained on Wikipedia with hyperlinks) and biomedical domain ( pretrained on PubMed with citation links), LinkBERT outperforms previous BERT models across a wide range of downstream tasks. The gains are notably large for multi-hop reasoning, multi-document understanding and few-shot question answering, suggesting that LinkBERT effectively internalizes salient knowledge through document links. Our results suggest that LinkBERT can be a strong pretrained LM to be applied to various knowledge-intensive tasks.

Reproducibility

Pretrained models, code and data are available at https://github.com/michiyasunaga/LinkBERT.
Experiments are available at https://worksheets.codalab.org/worksheets/0x7a6ab9c8d064a1d191335b270da2902e.

Acknowledgment

We thank Siddharth Karamcheti, members of the Stanford P-Lambda, SNAP and NLP groups, as well as our anonymous reviewers for valuable feedback. We gratefully acknowledge the support of NSF CAREER Award IIS-1552635; DARPA under Nos.
HR00112190039 (TAMI), N660011924033 (MCS); Funai Foundation Fellowship; Microsoft Research PhD Fellowship; ARO under Nos. W911NF-16-1-0342 (MURI), W911NF-16-1-0171 (DURIP); NSF under Nos. OAC-1835598 (CINES), OAC-1934578 (HDR), CCF-1918940 (Expeditions), IIS-2030477 (RAPID), NIH under No. R56LM013365; Stanford Data Science Initiative, Wu Tsai Neurosciences Institute, Chan Zuckerberg Biohub, Amazon, JPMorgan Chase, Docomo, Hitachi, Intel, KDDI, Toshiba, NEC, Juniper, and UnitedHealth Group.

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. Hitln: Hyper-text pre-training and prompting of language models. arXiv preprint arXiv:2107.06955.

David Ansari, Daniel Ansari, Roland Andersson, and Åke Andrén-Sandberg. 2015. Pancreatic cancer and thrombomembolic disease, 150 years after troussseau. Hepatobiliary surgery and nutrition, 4(5):325.

Akari Asai, Kazuma Hashimoto, Hannané Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In International Conference on Learning Representations (ICLR).

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högborg, Ulla Stenius, and Anna Korhonen. 2016. Automatic semantic classification of scientific literature according to the hallmarks of cancer. Bioinformatics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In Empirical Methods in Natural Language Processing (EMNLP).

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. In North American Chapter of the Association for Computational Linguistics (NAACL).

Rishi Bommasani, Drew A. Hudson, Elsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeanette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quiny Davis, Dora Demszky, Chris Donahue, Moussa Douniboua, Esin Durmus, Stefano Ermon, John Etchemendy, Kawan Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karon Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamchetti, Geoff Keeling, Fereshhteh Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kudluputi, Ananya Kumar, Faisal Ladak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Yuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mishandranik, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avaniya Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakovenko. 2013. Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems (NeurIPS).

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In Association for Computational Linguistics (ACL).

Alex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I Furlong. 2015. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. BMC bioinformatics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Saxtry, Amanda Askell, et al. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems (NeurIPS).

Avi Caciu, Arma Cohan, Iz Beltagy, Matthew E Peters, Arie Cattan, and Ido Dagan. 2021. Cross-document language modeling. arXiv preprint arXiv:2101.00406.

Iacer Calixto, Alessandro Raganato, and Tommaso Pasini. 2021. Wikipedia entities as rendezvous of document language modeling. arXiv preprint arXiv:2107.06955.
Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In International Conference on Learning Representations (ICLR).

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In Association for Computational Linguistics (ACL).

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. In Association for Computational Linguistics (ACL).

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In Machine Learning Challenges Workshop.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In North American Chapter of the Association for Computational Linguistics (NAACL).

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. Journal of biomedical informatics.

Matthew Dunn, Levent Sagun, Mike Higgins, V Uğur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. arXiv preprint arXiv:1704.05179.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In Workshop on Machine Reading for Question Answering.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In Association for Computational Linguistics (ACL).

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. arXiv preprint arXiv:2007.15779.

Kelvin Guu, Kenton Lee, Zora Tang, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In International Conference on Machine Learning (ICML).

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. Integrating graph contextualized knowledge into pre-trained language models. In Findings of EMNLP.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In International Conference on Learning Representations (ICLR).

Monika R Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. 2000. On near-uniform url sampling. Computer Networks.

María Herrera-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. Journal of biomedical informatics.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2020. Strategies for pre-training graph neural networks. In International Conference on Learning Representations (ICLR).

Di Jin, Eileen Pan, Nassim Outfattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. Applied Sciences.

Qiao Jin, Bhuvan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In Empirical Methods in Natural Language Processing (EMNLP).

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Association for Computational Linguistics (ACL).

Vladimir Karpukhin, Barlas Öğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval
Anita Khadka, Ivan Cantador, and Miriam Fernandez. 2020. Exploiting citation knowledge in personalised recommendation of recent scientific publications. In Language Resources and Evaluation Conference (LREC).

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hanannah Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In Findings of EMNLP.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at jmlpba. In Proceedings of the international joint workshop on natural language processing in biomedicine and its applications.

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martin Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, and Ander Itxaurrondo. 2017. Overview of the biocreative vi chemical-protein interaction track. In Proceedings of the sixth BioCreative challenge evaluation workshop.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Ilia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics (TACL).

Jinhuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2021. The inductive bias of in-context learning: Rethinking pretraining example design. arXiv preprint arXiv:2110.04541.

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020a. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In Proceedings of the 3rd Clinical Natural Language Processing Workshop.

Patrick Lewis, Ethan Perez, Aleksandra Pikutas, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Advances in Neural Information Processing Systems (NeurIPS).

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. Database.

Yinhuan 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.

Ilya Loschilov and Frank Hutter. 2019. Decoupled weight decay regularization. In International Conference on Learning Representations (ICLR).

Zhengyi Ma, Zicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021. Pre-training for ad-hoc retrieval: Hyperlink is also you need. In Conference on Information and Knowledge Management (CIKM).

Eric Margolis, Stephen Laurence, et al. 1999. Concepts: core readings. Mit Press.

Anastasios Nentidis, Konstantinos Bougiatiotis, Anastasia Krithara, and Georgios Palioras. 2019. Results of the seventh edition of the bioasq challenge. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfie Yang, Iain J Marshall, Ant Niknova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In Proceedings of the conference. Association for Computational Linguistics. Meeting.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unified open-domain question answering with structured and unstructured knowledge. arXiv preprint arXiv:2012.14610.

Franco Piovella, Luciano Crippa, Marisa Barone, S Viganò D’Angelo, Silvia Serafini, Laura Galli, Chiara Belmonte, and Armando D’Angelo. 2002. Normalization rates of compression ultrasonography in patients with a first episode of deep vein thrombosis of the lower limbs: association with recurrence and new thrombosis. Haematologica, 87(5):515–522.

Vahed Qazvinian and Dragomir R Radev. 2008. Scientific paper summarization using citation summary networks. In International Conference on Computational Linguistics (COLING).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research (JMLR).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In Empirical Methods in Natural Language Processing (EMNLP).
Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *International Conference on Computer Vision (ICCV)*.

### A Ethics, limitations and risks

We outline potential ethical issues with our work below. First, LinkBERT is trained on the same text corpora (e.g., Wikipedia, Books, PubMed) as in existing language models. Consequently, LinkBERT could reflect the same biases and toxic behaviors exhibited by language models, such as biases about race, gender, and other demographic attributes (Sheng et al., 2020).

Another source of ethical concern is the use of the MedQA-USMLE evaluation (Jin et al., 2021). While we find this clinical reasoning task to be an interesting testbed for LinkBERT and for multi-hop reasoning in general, we do not encourage users to use the current models for real world clinical prediction.

### B Fine-tuning details

We apply the following fine-tuning hyperparameters to all models, including the baselines.

**MRQA.** For all the extractive question answering datasets, we use $\text{max}_\text{seq}\_\text{length} = 384$ and a sliding window of size 128 if the lengths are longer than $\text{max}_\text{seq}\_\text{length}$.

For the -tiny scale (BERT$_\text{tiny}$, LinkBERT$_\text{tiny}$), we choose learning rates from $\{5e^{-5}, 1e^{-4}, 3e^{-4}\}$, batch sizes from $\{16, 32, 64\}$, and fine-tuning epochs from $\{5, 10\}$.

For -base (BERT$_\text{base}$, LinkBERT$_\text{base}$), we choose learning rates from $\{2e^{-5}, 3e^{-5}\}$, batch sizes from $\{12, 24\}$, and fine-tuning epochs from $\{2, 4\}$.

For -large (BERT$_\text{large}$, LinkBERT$_\text{large}$), we choose learning rates from $\{1e^{-5}, 2e^{-5}\}$, batch sizes from $\{16, 32\}$, and fine-tuning epochs from $\{2, 4\}$.

**GLUE.** We use $\text{max}_\text{seq}\_\text{length} = 128$.

For the -tiny scale (BERT$_\text{tiny}$, LinkBERT$_\text{tiny}$), we choose learning rates from $\{5e^{-5}, 1e^{-4}, 3e^{-4}\}$, batch sizes from $\{16, 32, 64\}$, and fine-tuning epochs from $\{5, 10\}$.

For -base and -large (BERT$_\text{base}$, LinkBERT$_\text{base}$, BERT$_\text{large}$, LinkBERT$_\text{large}$), we choose learning rates from $\{5e^{-6}, 1e^{-5}, 2e^{-5}, 3e^{-5}, 5e^{-5}\}$, batch sizes from $\{16, 32, 64\}$ and fine-tuning epochs from 3–10.

**BLURB.** We use $\text{max}_\text{seq}\_\text{length} = 512$ and choose learning rates from $\{1e^{-5}, 2e^{-5}, 3e^{-5}, 5e^{-5}, 6e^{-5}\}$, batch sizes from $\{16, 32, 64\}$ and fine-tuning epochs from 1–120.

**MedQA-USMLE.** We use $\text{max}_\text{seq}\_\text{length} = 512$ and choose learning rates from $\{1e^{-5}, 2e^{-5}, 3e^{-5}\}$, batch sizes from $\{16, 32, 64\}$ and fine-tuning epochs from 1–6.