Representing Long Documents with Contextualized Passage Embeddings

Anonymous authors

Abstract

In this study we investigated a method for processing a large document collection with many long documents. The goal was to improve the processing runtime and memory requirements for document level tasks. We propose to represent the document collection by a database of passage embeddings instead by the document’s tokens. For our approach the passage embeddings should be task-agnostic, such they can be pretrained, precomputed and then be reused for many tasks. We propose a pretraining method for passage embeddings with (i) a passage encoder (PE) coupled with (ii) a bidirectional document encoder (BDE) over the passage embeddings. BDE can be finetuned for different downstream tasks to quickly learn a new model for a new task. In experiments, we found that PE+BDE is competitive with token-level or sentence-level models and sometimes even better. For plagiarism detection, for example, we improve over a Longformer-based model by +14 accuracy points. For passage classification on the contract understanding task PE+BDE reaches an AUPR 60.3 while a token-level information extraction approach using RoBERTa-large obtained an AUPR of 48.2.

1. Introduction

Recently, the challenge of handling long and structured documents has started to receive attention in the NLP community. The challenge of processing many long documents is the high cost this incurs. Current state-of-the-art models —like BERT [Devlin et al., 2019]— can only efficiently process 512 tokens at a time. For a 100 page document this would yield roughly 200 document chunks which have to be processed independently, i.e. document semantics can be lost. There are recent improvements over BERT like the Longformer [Beltagy et al., 2020]. The largest Longformer-based model can chunk a 100 page document in only 6 parts a 16K tokens but would then require a GPU with 48 GB working memory. However, even with more efficient models that operate on the token level the processing cost is still bound by the number of tokens in the document.

In this study we investigate a scenario of a large document collection with many long documents like they exist for example in the scientific, technical and legal domain. We assume
that the document collection will be processed for many types of tasks, like information extraction and document classification. Also, we assume that not all tasks are known a priori such that task-specific preprocessing of the documents is not feasible. Our hypothesis is the processing cost can be reduced when parts of the computation can be cached.

Therefore—similar to state-of-the-art approaches in question answering—we propose to represent documents with passage embeddings. In our study a passage is a document chunk of a certain token size, which is obtained in an ad-hoc fashion, i.e. chunks of 128 sub-word tokens in the following. The underlying hypothesis is that passage embeddings can encode a sufficient amount of information for some tasks. Alternatively they could be used for a coarse pre-classification over many passages, such that only a small subset of a document’s passages is processed on a token-level. Processing documents with passage embeddings yields a reduction in run-time and memory complexity in the order of the token window size of the passage. Moreover, document collections in the scientific and legal domain consist of many long documents, yet they produce comparably only very little expert annotations. This motivates the investigation into self-supervised trained models for documents, to enable transfer learning from a pretrained model.

To obtain passage embeddings we use a passage encoder (PE) based on the Transformer [Vaswani et al., 2017]. To encode document-wide semantics among the passage embeddings and to be able to pretrain a model that is able to represent a document we propose a bidirectional document encoder (BDE) also based on the Transformer. The technical novelty is the task-agnostic pretraining strategy that enables self-supervised pre-training of PE and BDE on the document collection. For evaluating the approach we need document benchmarks and we assume that such a benchmark should test the computational efficiency for long documents and also require inference and shared semantics across the whole document. Therefore, we evaluate our approach on document pair classification and passage classification, and perform a small analysis on dense retrieval benchmarks. However, analyzing currently available document benchmarks we found that some do not contain long documents, i.e. documents of multiple pages, or are not challenging regarding the inference within the document. Therefore, we propose a new document pair classification benchmark WikiPairEntityMatch in which the task is to determine if in a pair of documents both do mention one or more common entities, i.e. cross document coreference.

The goal of this study was to investigate if precomputing passage embeddings and pretraining a bidirectional document encoder is a feasible approach in the setting of a large document collection with many long documents. While the improvement of this approach in terms of computational complexity is obvious, it is unclear what penalty it incurs on the task performance when there is no access to lexical information. Our experimental results on document pair classification indicate that the approach performs mostly only a bit lower than the best token-level method from a concurrent study based on the Longformer [Caciularu et al., 2021]. However, for the plagiarism detection benchmark—that contains the longest documents—our proposed approach even improved by +14% accuracy. For passage classification BDE achieved up to 60.4 AUPR on the passage-level while the best token-level model based on RoBERTa-large achieved 48.2 AUPR. We also show that there might be a benefit for dense retrieval, when the passage embeddings are contextualized by the document encoder. We conclude that our proposed approach is feasible and enables efficient sweeps over large document collections with long documents.
2. Contextualized embeddings and the bidirectional document encoder

In the introduction we motivated our approach of representing documents using precomputed embeddings of document chunks and pretraining a bi-directional document encoder. In this section we describe the setting and its challenges in more detail. Then we describe the architecture and motivate the design choices we made.

2.1 Overview

Figure 1 shows an overview of the approach that we propose and the three downstream tasks in which we evaluate it in: dense passage retrieval, document pair classification and passage classification. The architecture we use is a hierarchical Transformer similarly to [Pappagari et al., 2019, Zhang et al., 2019, Yang et al., 2020]. First, a lower BERT-style Transformer computes a dense vector representation of a token sequence, which we denote as the passage encoder (PE). Then, the sequence of dense passages are fed into another BERT-style Transformer, i.e. the bidirectional document encoder (BDE). Both, PE and BDE can be pretrained in a self-supervised way on a document collection. Then PE is used to precompute the passages of the document collection. BDE yields three different types of applications:

(i) Passage retrieval with contextualized passage embeddings: Instead of using the non-contextualized passages precomputed with PE like in prior work [Karpukhin et al., 2020], BDE can be used to create contextualized passages.

(ii) Document-level classification: The pretrained BDE can be finetuned for document-level classification, e.g. the document-pair classification task.

(iii) Passage classification: The pretrained BDE can be finetuned for passage classification.
3. Pretraining Passage Encoder and Bidirectional Document Encoder

In this section we describe the pretraining objectives for the passage encoder (PE) and the bidirectional document encoder (BDE).

3.1 Architecture and Pretraining Objective

See Figure 2 for an overview. The passage encoder (PE) receives a text passage $T = [t_1, t_2, ..., t_n]$ and computes a dense passage embedding (white rectangle) which is obtained from a special token like the CLS token in BERT. For pretraining PE we use another BERT-like Transformer which we call the passage decoder (PD). PD receives the passage embedding from PE together with the text passage $\hat{T} = \text{mask}(T)$ in which most of the tokens have been masked out. PD attempts to reconstruct the original text based on the passage embedding and the heavily masked out text. The bidirectional document encoder (BDE) receives a sequence of the document’s passage embeddings $P = [p_1, p_2, ..., p_m]$. BDE produces contextualized passage embeddings (striped rectangles). BDE’s training objective is to predict the original passage for a masked out passage position in the sequence of passages $\hat{P} = \text{mask}(P)$. Novel in our proposed approach are the two pretraining objectives:

**Passage reconstruction.** The goal for the training objective for the passage encoder is to obtain passage embeddings that contain information from the passage for many different tasks. Thus we use a passage decoder (PD) —a BERT-style Transformer— to reconstruct the original text passage from the passage embedding. Instead of a full reconstruction we only want to encourage the model to encode information from the passage that is the most specific, e.g. entities, relations and events. As a proxy for this, we mask out tokens based on the token frequency, i.e. all tokens except the most frequent tokens are masked out. This provides a template for the decoder which the model has to fill out using the passage embedding. For example, PE receives the original text "The 2015 Illinois Fighting Illinois football team represented the University of Illinois ..." and creates a passage embedding. Then PD receives the passage embedding and the masked out text "The _ _ _ team _ the University of _ ..." and has to reconstruct the original text passage. Thus the reconstruction objective is similar to one step of non-autoregressive decoding. The loss for this task is only

![Figure 2: Architecture and pretraining objectives for BDE.](image-url)
computed for PD and is the standard cross entropy loss over the vocabulary for classifying the masked out tokens. For training PE/PD all the vocabulary weights are shared between both encoders and decoders.

**Masked document modelling.** The document’s passage embeddings are fed to BDE. Here the objective is masked document modelling where masked out passages have to be predicted, i.e. similar to BERT’s masked language modelling objective. The target vocabulary is very large, i.e. Wikipedia with passages of 100 words yields $22M$ passages. Thus in practice we used a surrogate loss to predict the correct passage from the in-batch passages, i.e. the cross entropy loss for the classification over the in-batch passages for the masked out passages.

**Discussion.** In contrast to [Zhang et al., 2019] we split the objective of using the document context and reconstructing the token sequence into two objectives, and we frame the reconstruction in a template filling problem. The goal of [Zhang et al., 2019] was to pretrain their model for generation, while our goal is to pretrain the model for information extraction tasks.

4. **Experiments**

In the following experiments we investigated two distinct scenarios for which PE and BDE can be used in downstream tasks:

(i) *Use precomputed contextualized embeddings from BDE* (precompute-BDE) for a downstream task like passage retrieval. We are interested if the retrieval task can benefit from the document wide semantics that are encoded in the passage embeddings

(ii) *Use precomputed embeddings from PE and only finetune BDE* (precompute-PE+finetune-BDE) for a downstream task. This is motivated by a scenario with a large document collection. Therefore, one shared database of passage embeddings serves as representation of the document collection for downstream tasks that will be reused by many different tasks like document pair classification and passage classification. When precompute-PE+finetune-BDE reaches competitive performance to a token-level model then the dramatic speedup of using BDE would become attractive.

In the following sections we first describe how we pretrained PE and BDE. Then we describe the setup in the downstream tasks and report experimental results.

4.1 **Pretraining.**

The pretraining data was a current dump of English Wikipedia with approximately 6 million documents preprocessed into text. We selected 20,000 documents for validation and testing each. The passage size is a hyperparameter and we picked a size of 128 tokens with no overlap. This did yield approximately 36 million training passages.

For pretraining we initialized PE, PD and BDE from BERT base uncased. First we trained PE with the passage reconstruction loss. We chose a vocabulary cutoff for masking out input tokens for PD such that the 300 most frequent tokens are not masked out. We trained PE for approx. one epoch on 8xV100 GPUs, which takes roughly 1 day. We kept all default hyperparameters for BERT except the learning rate which was set to $1e^{-04}$. Then we jointly trained PE/PD with BDE with the additional masked document modelling
Table 1: Results (best, second best) for passage retrieval on TriviaQA (TRIVIA) and Natural Questions (NQ). We report the in-batch MRR on the validation data from KILT (the test data is not public). Contextualized NO is the baseline setting in which the query encoder retrieves answer passages using the non contextualized PE embeddings. Our proposed approach is using the contextualized embeddings from BDE, i.e. Contextualized YES. The setting in which PE is not finetuned simulates a fixed document representation for which a new QA task is learned.

4.2 Passage retrieval task

With this task we want to investigate if it is beneficial for retrieval when the passage embeddings are contextualized with BDE. In the retrieval task the goal is given a natural language query to retrieve relevant passages from a knowledge source.

Data. We based our experiments on KILT [Petroni et al., 2020] which is a benchmark based on Wikipedia as a knowledge source for knowledge intensive tasks that consists of eleven datasets spanning five tasks, including open domain question answering (QA). We analyzed the QA datasets in KILT w.r.t. how important it is to understand the whole document for a question. As a proxy for this we computed statistics about the location of the answer in the document, with the notion that passages later in the document tend to be more dependent on information in passages prior to them. The three relevant QA datasets were Natural Questions [Kwiatkowski et al., 2019], HotpotQA [Yang et al., 2018], TriviaQA [Joshi et al., 2017] as well as two slot filling datasets TREx [Elsahar et al., 2018] and Zero Shot RE [Levy et al., 2017]. We computed statistics over the location of the provenance of the query’s answer which is one of the meta-data available in KILT and found that — except for Natural Questions and Trivia QA — over 90% of the questions in most datasets can be answered from the first passage in the document. Thus, HotpotQA, TREx and Zero Shot RE would not benefit from a contextualization of the passages. However, the Natural Questions and TriviaQA benchmark contain the answer’s provenance in the 7th
<table>
<thead>
<tr>
<th>Document Pair Task</th>
<th>Avg. #Pass</th>
<th>#Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAN citation</td>
<td>1.72</td>
<td>100K</td>
</tr>
<tr>
<td>S2ORC citation</td>
<td>3.08</td>
<td>150K</td>
</tr>
<tr>
<td>OC citation</td>
<td>2.44</td>
<td>240K</td>
</tr>
<tr>
<td>PLA plagiarism</td>
<td>33.46</td>
<td>18K</td>
</tr>
<tr>
<td>WikiPairEntityMatch mention same entity</td>
<td>7.00</td>
<td>380K</td>
</tr>
</tbody>
</table>

Table 2: Dataset statistics for the document pair classification task. AAN, S2ORC, OC and PLA were published by Zhou et al. [2020]. We propose WikiPairEntityMatch which adds a new task and contains many long documents.

passage or later in the Wikipedia article in 10% and 20% of the questions respectively. Thus we focussed our experiments on TriviaQA and Natural Questions.

**Model.** For this task an additional query encoder (QE) is trained to compute a query embedding from a natural language query. QE is initialized from the pretrained PE and is trained with a cross entropy loss for the classification objective to predict the answer passage out of all in-batch passages. This is similar to the objective used by [Karpukhin et al., 2020]. To investigate if it is beneficial when the passage embeddings are contextualized with BDE we compare two settings: (i) The baseline is training and evaluating QA against the document passage embeddings obtained with PE, which are not contextualized. (ii) The Training and evaluating QA against the document passage embeddings obtained with BDE, which are contextualized.

**Experimental settings.** PE/BDE was initialized from our checkpoint pretrained on Wikipedia. We trained PE/BDE with a batch size of 4 documents on 4 GPUs with gradient accumulation over 30 steps, which yields a total batch size of 480 per update step. Furthermore, we restrict a document to a maximum size of 32 passages which covers 95% of the relevant Wikipedia articles. We report the in-batch MRR, i.e. the ranking of the correct answer passage against all in-batch passages. For evaluation we choose a batch size of 32 which yields an average of roughly 700 passages per batch to rank the correct passage against.

**Results** See Table 1 for the results. Comparing the results for retrieving contextualized with non-contextualized passages we observed that contextualizing helped, as we can see an increase in MRR between them. Furthermore, comparing the results when PE is finetuned or not we observe that there is no consistent gain when PE is further adapted for the QA task.

4.3 Document pair classification

In this task, given a pair of documents the goal is to classify the pair w.r.t. a given task.

**Model.** In document pair classification we construct a BERT-style "sentence" of two documents, i.e. [CLS] [SEQ] PE₁₁ PE₁₂ ... [SEQ] PE₂₁ PE₂₂. This sentence is the input to
Table 3: Results (best, second best) for document pair classification experiments. We compare the following settings: Finetune Enc. denotes if the encoder (PE in our approach) was finetuned during training. Level denotes which level of access the model has during inference. CD-LM is a Longformer based pretrained language model for cross document classification [Caciularu et al., 2021]. BERT-AVG and BERT-HAN and GRU-HAN are the results reported by Zhou et al. [2020]. Model initializations/pretraining: BDE BERT was initialized from vanilla BERT-base-uncased; BDE WIKI was pretrained on Wikipedia; BDE TRAIN was pretrained with the PE/BDE pretraining on the joint training data of all the tasks.

BDE, see also Figure 1. The document pair classification task uses then the CLS embedding to make a prediction with an additional classifier head.

Data. We use the three citation recommendation benchmarks and plagiarism detection benchmark as proposed by Zhou et al. [2020]. We propose WikiPairEntityMatch as a new benchmark to add another difficult challenge to the range of benchmarks for long documents. The benchmark was constructed by using pairs of Wikipedia articles that mention the same entity (which we can extract via the intra Wikipedia links) as positive examples. As negative examples we sample 10 pairs of Wikipedia articles that have an overlap in their name but have zero overlap w.r.t intra Wikipedia links. The validation and test data are each of size 12K and the test data is either sampled such that the Wikipedia pages and some entities in them are unseen or seen (IID). See Table 2 for an overview of dataset statistics.

Experimental settings. PE/BDE was either initialized from our checkpoint from Wikipedia and once pretrained on the collective training data of all tasks. We finetuned PE/BDE with a batch size of 4 documents with gradient accumulation over 80 steps. We compare against the reported F1 results of the original dataset authors and with concurrent work in comparable settings.

Results. For most tasks and most settings BDE is competitive and achieves either the second best or best result. The only setting in which our approach does not yield competitive
Table 4: Results for the passage classification experiment. The model BDE BERT was initialized from vanilla BERT-base-uncased. BDE TRAIN was pretrained with the BDE pretraining procedure on the training data of the task. RoBERTa are the best results reported by [Hendrycks et al., 2021]. They were obtained with a SQUAD-style QA model, therefore the metrics are not directly comparable to ours.

results is when PE is initialized from pretrained Wikipedia and is not finetuned during training. In all other settings BDE outperforms the models from Zhou et al. [2020] and is competitive to the Longformer-based model by [Caciularu et al., 2021].

4.4 Passage classification.

In this task a passage has to be classified into one or more categories. Such a task is best exemplified by the dataset that we used in this experiment. The contract understanding dataset (CUAD) [Hendrycks et al., 2021], contains 500 expertly annotated documents from which more than 50% are longer than 50 passages and more than 25% are longer than 100 passages. This means that processing such a dataset on the token level for exploration and training classifiers is challenging. In fact the authors of the dataset had to resort to a sliding window approach to train models and make inference on this dataset. The annotations are also only meant to be highlights of sections that an expert should review, thus it is very reasonable to assume that such a task can be performed with a reasonable granularity on the passage level as well.

Data. The data are contract documents with span level annotations for certain slot fillers, such as “Effective Date” or “Renewal Term”. They have been annotated by lawyers and cover 500 documents with up to 100 pages.

Model. In document pair classification we construct a BERT-style ”sentence” of one document, i.e. [CLS] [SEQ] PE_{11} PE_{12} … . This sentence is the input to BDE, see also Figure 1. The passage sequence classification task predicts labels for each passage with an additional classifier head. In this particular dataset it is possible that one passage could cover multiple annotations and thus it is modelled as a multilabel task.

Experimental settings. PE/BDE was either initialized from our checkpoint pretrained on Wikipedia, or it was pretrained with the BDE/PE pretraining objective on the respective training data and PE then kept fixed during finetuning BDE. We trained PE/BDE with a batch size of 1 document with gradient accumulation over 80 steps. We set the maximum number of passages per document to 100. We report F1 and the Area under Precision Recall Curve (AURP). We compare against the dataset’s authors’ reported results.
Results. The results in Table 4 show that the proposed approach in BDE is very competitive, although as we mention also in the results table the results have been obtained very differently and our model only yields a passage level prediction.

5. Related Work

Transformer for large sequences The current state of the art NLP approaches use Transformer based bidirectional pretrained language models (BIPLM), which have quadratic complexity in the number of input tokens. Therefore, their input is usually limited to 512 tokens. [Beltagy et al., 2020, Zaheer et al., 2020] studied how to reduce the runtime complexity and effectively increased the input size for BIPLMs to 4096 tokens or more.

Passage embeddings and passage retrieval. The method to retrieve relevant passages from many long documents is usually exploiting token overlap between the query and the passages. In QA the goal is open domain question answering in which the collection of evidence documents is the whole Wikipedia from which relevant passages have to be retrieved. TF-IDF based retrieval — which exploits token overlap between the query and passages — has shown to be a bottleneck in QA, and can be improved by retrieving relevant passages with dense passage embeddings [Lee et al., 2019, Karpukhin et al., 2020, Asai et al., 2020, Logeswaran et al., 2019, Wu et al., 2020, Guu et al., 2020, Lewis et al., 2020, eli, hay]. [Karpukhin et al., 2020] and Lee et al. [2019] trained a BERT based model to compute a dense passage representations, such that relevant passages are classified higher given a query than irrelevant passages. [Karpukhin et al., 2020] reported that using fixed-length passages performs better than natural paragraphs in their preliminary trials in both retrieval and final QA accuracy. Most notably, Lee et al. [2019] add the Wikipedia page title to each passage, which they found to be helpful, which is an indicator that supports our hypothesis.

Hierarchical Transformer [Pappagari et al., 2019] used BERT encoded passages fed into yet another Transformer or RNN to perform document classification, which they reported to performing better than other approaches. [Yang et al., 2020] proposed to hierarchically encode documents with stacked Transformers to encode passages and finally produce a single document vector embedding, which they successfully use for document matching. In [Zhang et al., 2019] they compute sentence embeddings over which yet another Transformer performs abstractive summarization which outperforms other approaches.

6. Conclusion

In conclusion we showed that performing tasks on a passage level seems to be a feasible and competitive approach to token level approaches. We also showed that contextualizing passages can improve results such that the document-wide semantics can be recovered. We presented our approach for pretraining and then the experimental results on three downstream tasks. We will release the code, our newly proposed WikiPairEntityMatch benchmark and pretrained models to foster research for pretraining passage embeddings and pretrained and finetuning BDE-like models.
References


Anonymous authors


Representing Long Documents with Contextualized Passage Embeddings


