DocSplit: Simple Contrastive Pretraining for Large Document Embeddings

Anonymous EMNLP submission

Abstract

Existing model pretraining methods only consider local information. For example, in the popular token masking strategy, the words closer to the masked token are more important for prediction than words far away. This results in pretrained models that generate high-quality sentence embeddings, but low-quality embeddings for large documents. We propose a new pretraining method called DOCSPLIT which forces models to consider the entire global context of a large document. Our method uses a contrastive loss where the positive examples are randomly sampled sections of the input document, and negative examples are randomly sampled sections of unrelated documents. Like previous pretraining methods, DOCSPLIT is fully unsupervised, easy to implement, and can be used to pretrain any model architecture. Our experiments show that DOCSPLIT outperforms other pretraining methods for document classification, few shot learning, and information retrieval tasks.

1 Introduction

Generating high-quality text embeddings for documents is a long-standing open problem. Most previous studies focus on either learning sentence-level representations (**???**) where training data usually contain short text or designing specific model structures for larger-range dependencies (**??**), but effective and efficient document representation learning methods are less explored.

In this paper, we present the DOCSPLIT which is the first unsupervised pretraining method designed specifically for large documents. The training procedure of DOCSPLIT can work with any model architecture to improve document representations. Specifically, DOCSPLIT uses contrastive learning, and our key contribution is a new method for generating positive samples for contrastive learning.

To evaluate the quality of document embeddings, we conduct standard and few-shot text classification on five large document datasets involved in News and scientific articles. Moreover, to further validate the effectiveness of our method, we also conduct document retrieval on the AAN dataset which is designed for long document understanding evaluation. The experimental results show that DOCSPLIT with two kinds of model structures (i.e., BERT and Longformer) can both achieve significant improvements compared to state-of-the-art baselines.

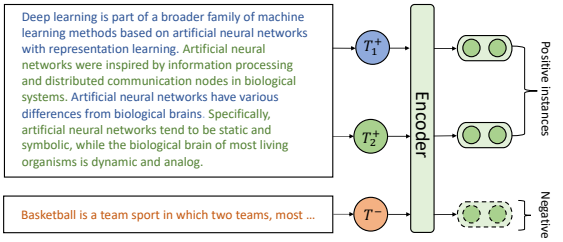

Figure 1: Document splitting (DOCSPLIT) is a new pretraining strategy that uses contrastive learning. The hard part of contrastive learning is generating positive instances. DOCSPLIT generates these pairs from an input data point by randomly assigning each sentence in the input document to one of the pairs.

Our paper is organized as follows. In Section 2 we formally define the contrastive learning problem and our novel DOCSPLIT training method. In Section **??** we develop a new experimental evaluation procedure for documents. We conclude in Section **??** by emphasizing that all of our models and datasets are open source.

2 Method

In this section, we first formally define contrastive learning, then we describe our DOCSPLIT method.

2.1 Contrastive Learning

Contrastive Learning aims to learn effective representations by pulling semantically close neighbors together and pushing apart non-neighbors in the

latent space (**?**). It assumes a contrastive instance $\{x, x^+, x_1^-, \ldots, x_{N-1}^-\}$ including one positive and $N-1$ negative instances and their representations $\{\mathbf{h}, \mathbf{h}^+, \mathbf{h}_1^-, \ldots, \mathbf{h}_{N-1}^-\}$, where x and x^+ are semantically related. we follow the contrastive learning framework (**??**) and take cross-entropy as our objective function:

$$l = -\log \frac{e^{\text{sim}(\mathbf{h},\mathbf{h}^+)/\tau}}{e^{\text{sim}(\mathbf{h},\mathbf{h}^+)/\tau} + \sum_{i=1}^{N-1} e^{\text{sim}(\mathbf{h},\mathbf{h}_i^-)/\tau}} \quad (1)$$

where τ is a temperature hyperparameter and $\text{sim}(\mathbf{h}_1, \mathbf{h}_2)$ is the cosine similarity $\frac{\mathbf{h}_1^\top \mathbf{h}_2}{\|\mathbf{h}_1\| \cdot \|\mathbf{h}_2\|}$.

For negative instances, we use in-batch instances following previous contrastive frameworks (**??**). The critical problem in contrastive learning is how to construct positive pairs (x, x^+).

2.2 Document Splitting (DocSplit)

Our main contribution is the DocSplit method to construct positive instances for documents. The idea is simple to describe and implement. Given an input document, we first split the document into sentences. Then each sentence is randomly assigned to eith

There are several intuitive reasons why we should expect DocSplit to generate good document pairs. First, we can think of each positive instance as a summary of the original document. Second, our splitting strategy captures global information about the document.

3 Related Work

There are two categories of related work: models trained using contrastive learning, and models designed for large documents.

3.1 Contrastive Pretraining

Contrastive learning has shown remarkable recent success for developing sentence embeddings. The simplest method is SimCSE (**?**), which uses dropout to generate correlated positive samples. The contrastive tension method (**?**) is similarly generic, but has a much more complicated implementation involving multiple models trained jointly. Because both of these pretraining strategies are generic, they can be used with any type of input including documents; but they do not take explicit advantage of document structure. The INSTRUCTOR model (**?**) is the current state-of-the-art model for most downstream tasks. The contrastive objective for this model requires a specially constructed corpus of manual-human annotations, and this corpus is limited only to sentence-level annotations instead of document-level annotations. We show that our model significantly improves on INSTRUCTOR on document level tasks, and it is not clear how to extend the INSTRUCTOR model to document-level tasks because human annotation for documents is significantly more expensive than for sentences. The Contriever model (**?**) uses a contrastive objective most similar to our own. They use the inverse cloze and document cropping tasks for pretraining. In document cropping, a document is divided in half and the two halves are used as the positive samples; in inverse cloze, a contiguous substring of the document is used as one positive sample and all other strings are used as the negative sample. The DocSplit pretraining method can be seen as a generalization of these methods.

3.2 Large Document Architectures

All of the models discussed in Section 3.1 above are based off of the BERT architecture (**?**). This architecture uses an attention mechanism that requires $O(n^2)$ memory and runtime, where n is the size of the attention window. The maximum size of a document that a model can understand is limited by this window size, and so compute for these models scales quadratically with the length of the documents.

A growing body of research focuses on developing new architectures with reduced computational requirements that enable processing larger documents. The LongFormer (**?**) and BigBird (**?**) models pioneered this line of research, and both models reduce the runtime of the attention mechanism to $O(n)$. A variety of other architectures have subsequently been proposed (e.g. **?????**). **?** provide a survey of this large body of work. Importantly, all of this research focuses only on improving the computational aspects of model architecture, and none of these models use a training objective designed specifically for large documents. Because the DocSplit pretraining method is model agnostic, we can easily apply it to any of these newly proposed model architectures. For computational reasons, we limit our experimental comparisons in Section **??** below to the LongFormer and BigBird models since these are the two most influential model architectures designed for large documents. We find a large performance improvement when these models are trained with DocSplit, and ex-

pect this performance improvement would extend to similar models as well.

4 Experiments

We perform a careful ablation study to isolate the effects of the DocSplit pretraining method on downstream task performance. First, we pretrain separate models for each group of baseline models described in Section 3 above. Then, we perform downstream experiments on standard classification, few-shot learning, and information retrieval tasks. In all cases, our pretrained models significantly outperform prior work.

4.1 Pretraining Details

We pretrain two models on two different architectures. All prior work using contrastive learning discussed in Section 3.1 above evaluates their pretraining methods on the BERT architecture. To fairly compare against these methods, we pretrain our DocSplit$_{\text{bert}}$ model also on this architecture. Ultimately, however, we are interested in large document performance, and so we expect that a model architecture designed specifically for large documents will improve performance. We therefore also pretrain the DocSplit$_{\text{long}}$ model on the LongFormer architecture. This second model will be used to evaluate against other models designed specifically for large documents.

To pretrain both models, we follow the standard pretraining procedure for contrastive losses established by **?** and **?**. We simultaneously optimize both the masked language model (MLM) loss (with weight$= 0.1$) and the contrastive loss (with temperature $\tau = 0.05$). We use English Wikipedia articles as our pretraining dataset. These articles are long, and so we expect that a pretraining procedure designed for large documents will improve performance. The total number of training instances is 6,218,825. We use AdamW (**?**) with a learning rate of 5e-5. DocSplit$_{\text{bert}}$uses a batch size of 36. And due to the larger memory requirements of the LongFormer architecture, DocSplit$_{\text{long}}$uses a batchsize of 12. For both models, we know that performance improvements on downstream tasks must be due to the pretraining procedure and not the dataset because all baseline models include English language wikipedia in their training set.

4.2 Experiment 1: Text Classification

We fine tune DocSplit$_{\text{bert}}$, DocSplit$_{\text{long}}$, and all baseline models on five standard document datasets. The datasets are summarized in the table below:[1]

Dataset	Num Docs	Classes	Words / Doc		99th	95th	
			Mean	Max			
FakeNews	8,558,957	15	467	33,936	2,949	1,403	1
arXiv	2,162,833	38	138	925	289	255	
20News	18,846	20	258	11,554	1,865	714	
NYT	13,081	5	650	5503	1,264	1,043	
BBCNews	2,225	5	133	445	311	234	

Notice in particular that every dataset has documents that exceed the standard context window length of the BERT model. For these documents, we follow the standard procedure and truncate the documents before passing them to BERT-based models.

Table **??** shows the accuracy and F1 score of every model on these datasets. Notice that DocSplit$_{\text{bert}}$ out performs all BERT-based models discussed in Section 3.1, and DocSplit$_{\text{long}}$ outperforms all models on every dataset. There are no results for the INSTRUCTOR model on the 20News dataset because INSTRUCTOR was pretrained on this dataset and the authors state that evaluating INSTRUCTOR on datasets it was pretrained is incorrect due to data contamination.

4.3 Experiment 2: Few-shot Learning

Next we evaluate how DocSplit pretraining performs on classification tasks with a small number of training examples. We follow the standard procedure of artificially limiting the number of training examples used during training, and evaluating on the same test set. Figure 2 shows the classification accuracy on the 20News dataset as we vary the size of the training set. We see that DocSplit$_{\text{bert}}$ outperforms all other BERT-based models accross all sample sizes.

The results on other datasets and for LongFormer-based models are similar. A

[1] Citations for the datasets are: Fake News Corpus https://github.com/several27/FakeNewsCorpus; arXiv articles dataset https://www.kaggle.com/datasets/Cornell-University/arxiv; 20NewsGroups (**?**); New York Times Annotated Corpus (NYT) (**?**); and BBCNews http://mlg.ucd.ie/datasets/bbc.html.

Datasets	FakeNews		20News		arXiv		NYT		BBCNews	
Metrics	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BERT	54.98	42.17	62.34	54.19	68.52	20.46	95.11	92.65	91.06	90.34
CT-BERT	55.19	42.53	65.76	63.37	71.61	26.09	95.69	91.59	90.32	88.87
SimCSE	58.48	47.46	74.02	72.57	74.46	30.01	97.17	94.69	94.12	93.86
$\text{SimCSE}_{\text{long}}$	58.37	47.56	73.51	72.05	73.16	29.41	97.25	93.83	94.22	94.30
Contriever	58.21	47.25	75.86	74.28	75.35	28.24	96.94	92.71	94.66	94.57
INSTRUCTOR	59.26	47.92	–	–	75.52	30.46	97.06	93.66	95.19	95.16
$\text{DocSplit}_{\text{bert}}$	**60.04**	**50.14**	**76.89**	**74.85**	**76.66**	**32.24**	**98.20**	**96.05**	**95.56**	**95.58**
LongFormer	65.72	57.66	73.69	72.47	71.66	25.92	94.36	88.39	96.33	94.75
BigBird	57.44	47.87	70.35	68.91	71.58	25.05	97.13	94.33	94.11	94.62
$\text{DocSplit}_{\text{long}}$	**71.60**	**61.66**	**75.44**	**74.38**	**77.68**	**33.26**	**97.90**	**95.43**	**96.67**	**95.91**

Table 1: In the text classification of Experiment 1, models pretrained with DocSplit outperform baseline models in all cases. Larger numbers are better.

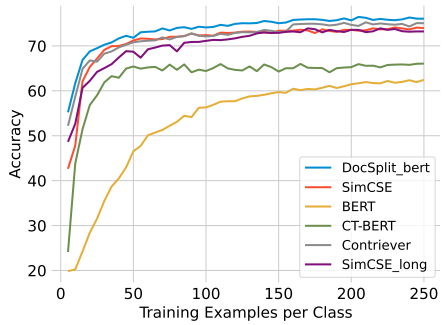

Figure 2: $\text{DocSplit}_{\text{bert}}$ outperforms all other BERT-based models in a few-shot classification task on the 20News dataset. FIXME: change x-axis to "training examples per class"; change DocSplit to $\text{DocSplit}_{\text{bert}}$, you can export the figure in tex instead of pdf to get the latex support needed for the legend

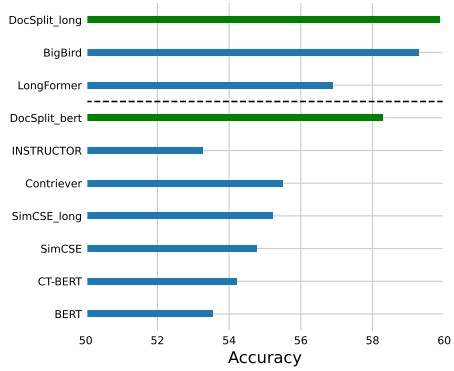

Figure 3

full set of results on other datasets is available in the Appendix.

4.4 Experiment 3: Document Retrieval

We conduct document retrieval to evaluate the ability to learn the similarity score between two vectors (**?**) of documents. We follow the document retrieval experiment (**?**) and use the ACL Anthology Network (AAN) (**?**) dataset, which identifies if two papers have a citation link, a common setup used in long-form document matching (**??**). Specifically, for all baselines, we use models to obtain document embeddings and finetune an MLP layer on two concatenated embeddings to predict if two documents have a citation link.

Table **??** shows the results of the document retrieval experiment on AAN dataset. We can find that our method achieves the best results due to the effectiveness of our pretraining methods based on document splitting.

5 Conclusion

In this work, we propose an unsupervised contrastive learning framework for large document embeddings. Our paper provides a new method for large document data augmentation without any supervision and language models can get large-scale pretraining on any large documents. We conduct extensive experiments on text classification tasks under fully supervised and few-shot settings and a document retrieval task for further evaluation. Results show that our pre-trained model greatly outperforms state-of-the-art text embeddings, especially when the training data is limited.

Limitations

The limitations of our method are as follows:

1. Generative models cannot be trained using DocSplit, but they can be trained using token masking.
2. We do not know how DocSplit will generalize to extremely large documents (e.g. book length). We lack the computational resources to train models on these larger documents.

Ethics Statement

Learning embeddings is a standard problem in natural language processing. Our approach uses standard training datasets and training procedures. There are therefore no direct ethical concerns with this research.

Our total compute is relatively small. Pretraining our DOCSPLIT$_{\text{bert}}$ and DOCSPLIT$_{\text{long}}$ models took about 1 month each on a standard desktop system with an NVidia 2080 GPU. Other models use larger GPU clusters that require considerably more energy. The finetuning procedures run in less than 1 day on the same system.

A Appendix

A.1 Training Details

For text classification, the learning rate for fine-tuning is 3e-4; the batch size is 8; the maximum sequence length is 512 tokens. We fine-tune the last MLP layer on these five datasets and evaluate the classification performance with accuracy and macro-F1 scores. For few-shot text classification, we sample 10 data instances per class for the FakeNewsCorpus dataset and the arXiv dataset and 5 data instances per class for the other three datasets. Other settings are the same as the standard text classification. Since there is randomness in sampling, we repeat every experiment 10 times and take the average value of metrics.