SimDE: Simple Contrastive Learning for Document Embeddings

Anonymous EMNLP submission

Abstract

Existing pretrained transformer models generate high-quality sentence and paragraph embeddings, but do not work well for large documents. A growing body of research tackles this problem by proposing new model architectures that are more efficient with large context windows, but all of this research still uses the classic token masking strategy for training. Token masking is a fundamentally *local* operation in that context of the entire document is rarely needed to correctly predict the masked token—the words much closer to the token will have a much larger influence. We propose a novel *global* method for training that requires a model to equally consider all portions of a document. Our Simple Document Embedding (SimDE) method is based on contrastive learning. Like token masking, SimDE is fully unsupervised and can work with any existing model architecture. We experimentally evaluate the SimDE on a wide range of architectures on standard classification datasets for large documents. We find an average improvement of 3.9% macro F1 score in a standard regime with large training datasets and a massive 12.0% improvement in the macro F1 score in a few-shot setting.

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 SIMDE which is the first unsupervised training method designed specifically for documents. The training procedure of SIMDE can work with any model architecture to improve document representations. Specifically, SIMDE uses contrastive learning, and our

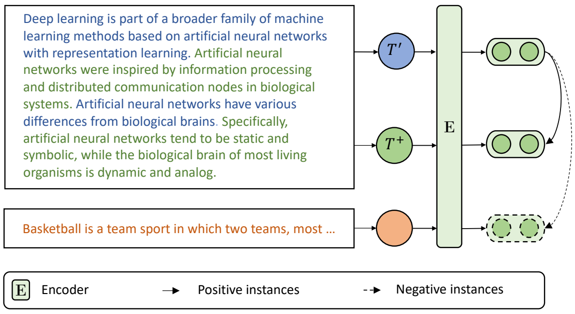

Figure 1: Overall framework of SIMDE. The document is randomly divided into two exclusive subsets of sentences and the two subsets work as positive pairs for contrastive learning. Other instances in the same batch are used as negatives.

key contribution is a new method for generating positive samples for contrastive learning. To this end, we first investigate the information redundancy with two methods (details in Appendix **??**) on five datasets for different lengths of documents. We find (1) information redundancy is larger as the length of the documents is increasing and (2) sentences from the same document usually contain repeated information. Based on this observation, we can assume that the model can still learn the semantics of documents even if we drop some sentences. Hence, as shown in Figure **??**, we randomly divide the original documents into two parts by sentences as positive pairs. Due to redundancies, our model can still recognize the two pairs have the same semantics. The intuition behind this method is that we expect the model will pull representations of two subsets together in the latent space by paying more attention to common keywords so that the model can learn key information from documents automatically.

To evaluate the quality of document embeddings, we conduct standard and few-shot text classification on five long text datasets involved in News and scientific articles. The experimental results show that SIMDE with two kinds of model structures (i.e., BERT and Longformer) can both achieve significant improvements compared to state-of-the-art baselines.

Our paper is organized as follows. In Section **??** we formally define the contrastive learning problem and our novel SIMDE 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 SIMDE 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:

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

where τ is a temperature hyperparameter and $\mathrm{sim}(\mathbf{h}_1, \mathbf{h}_2)$ is the cosine similarity $\frac{\mathbf{h}_1^\top \mathbf{h}_2}{\|\mathbf{h}_1\| \cdot \|\mathbf{h}_2\|}$. In this work, we encode input texts using a pre-trained language model such as BERT (**?**). Following BERT, we use the first special token `[CLS]` as the representation of the input and fine-tune all the parameters using the contrastive learning objective in Equation **??**.

2.2 SimDE

The critical problem in contrastive learning is how to construct positive pairs (x, x^+). In representation learning for visual tasks (**?**), an effective solution is to take two random transformations of the same image (e.g., flipping, rotation). Similarly, in language representations, previous works (**????**) apply augmentation techniques such as dropout, word deletion, reordering, and masking.

In this paper, we propose a new method to construct positive instances for documents. The basic idea of positive instance construction for contrastive learning is adding random noises to the original data for augmentation. The augmented data should have similar representations to the original data. Models trained by contrastive losses on augmented data will have an increased ability to learn important features in the data. To add random noises in documents, we find documents usually has higher information redundancy than sentences (Table **??** in Appendix). With this observation, we can have an assumption: the semantics of a document will not be changed even if we drop half of the document. We can construct positive pairs under this assumption easily on any text dataset without supervision. Specifically, for each document in the dataset, we randomly split sentences in the document into two subsets and the two sentence sets do not have intersections. In the two subsets, we keep the order of sentences in the original document to form two new documents. According to our assumption, the two new documents should have the same semantics and hence they are used as a positive pair in contrastive learning.

Consider an example (in Figure **??**) to understand our positive instance construction process: Suppose we have a document $T = (s_1, s_2, \ldots, s_n)$ where s_i is the i-th sentence in document and n is the number of sentences, each sentence will be sent to anchor set or positive set with the same probability (50%). The sentences in the same set (i.e., anchor or positive) will be concatenated in the same order of T to form one positive pair (T_1^+, T_2^+) for contrastive learning. Positive pairs constructed by this method will not contain the same sentence and hence prevent models from overfitting on recognizing the same sentences. Instead, models are guided to learn keywords appearing in positive instances so as to improve the ability to recognize key information. We split the document at sentence level instead of word level (e.g., word deletion for augmentation) because the word-level splitting will cause the discrepancy between pretraining and finetuning and then lead to performance decay.

For negative instances, we use in-batch instances following previous contrastive frameworks (**??**).

3 Experiments

In this section, we evaluate the effectiveness of our method by conducting text classification tasks. To eliminate the influence of different model structures and focus on the quality of text embeddings. We

Datasets	Data Size	Classes	Ave.	Med.
FakeNews	8,558,957	15	467	299
20News	18,846	20	258	153
arXiv	2,162,833	38	138	131
NYT	13,081	5	650	683
BBCNews	2,225	5	133	130

Table 1: Statistics of datasets. Ave. and Med. stand for the average and median number of words respectively in one data instance.

freeze the parameters of different text encoders and fine-tune only a multi-layer perceptron (MLP) to classify the embeddings of text encoders. We also visualize the attention weights between baselines and SIMDE.

3.1 Pretraining Details

For pre-training, we start from the pretrained BERT-BASE model (?) and the Longformer (?) model [1] We follow previous works (??): the masked language model (MLM) loss and the contrastive learning loss are used concurrently with in-batch negatives. We use English Wikipedia [2] articles as pretraining data and each article is viewed as one training instance. The total number of training instances is 6,218,825. Our pretraining learning rate is 5e-5, batch size is 36 and 12 for BERT and Longformer structure respectively. Our model is optimized by AdamW (?) in 1 epoch. The temperature τ in the contrastive loss is set to 0.05 and the weight of MLM is set to 0.1 following previous work (?).

3.2 Datasets

We use the following classic documents datasets to evaluate our method: (1) Fake News Corpus [3]; (2) 20NewsGroups (?); (3) arXiv articles dataset [4]; (4) New York Times Annotated Corpus (NYT) (?); and (5) BBCNews [5]. We do not use semantic textual similarity (STS) tasks (?) because the sentences in these tasks are short which is not suitable to evaluate long text embeddings.

3.3 Baselines

We compare our pre-trained model to the baselines of two groups. (1) BERT based models include BERT (?), SimCSE (?), CT-BERT (?). For a fair comparison, we also train a SimCSE with our pretraining dataset (SimCSE$_{\text{long}}$). (2) Transformers specified for long sequences include Longformer (?) and BigBird (?). We train two versions of SIMDE with BERT and Longformer (i.e., SIMDE$_{\text{bert}}$ and SIMDE$_{\text{long}}$) for comparison. We do not include RoBERTa (?) and IS-BERT (?) as our baselines because SimCSE achieves better results than these methods according to the paper.

3.4 Text Classification

In the standard text classification task, we classify text embeddings with the full training set. Training details are in Appendix **??**.

Results. Table **??** shows the evaluation results on different datasets. Overall, we can see that SIMDE achieves the best performance over the 5 document datasets and consistently improves the document embeddings with BERT and Longformer structures. Specifically, methods pretrained with contrastive objectives (i.e., CT-BERT, SimCSE) outperform general language representations (i.e., BERT) which indicates contrastive objectives designed for text embeddings can largely improve the ability of language models to produce high-quality text embeddings. SimCSE pretrained with our document data (i.e., SimCSE$_{\text{long}}$) has similar results as the original SimCSE which indicates simplely increasing the length of pretraining text cannot improve document embeddings. Compared to SimCSE and Longformer, our model achieves 3.9% and 9.4% average macro-F1 improvements with BERT and Longformer structures respectively. Hence, our contrastive learning method is effective for document embeddings.

3.5 Few-shot Text Classification

To show the performance of different text embeddings under low-resource settings, we evaluate our model with few-shot training instances. Training details are in Appendix **??**.

Results. Table **??** shows the results of few-shot text classification on these five datasets. We can see that SIMDE (i.e., SIMDE$_{\text{bert}}$ and SIMDE$_{\text{long}}$) achieves 12.0% and 24.3% macro-F1 improvements compared to SimCSE and Longformer respectively. These improvements are higher than standard text classification. Besides, we also compare the performance of different baselines and SIMDE$_{\text{bert}}$ with different numbers of training instances on 20News. The results in Figure **??** show

[1] The Longformer checkpoint is pretrained on long documents by MLM task and is available from Huggingface.

[2] https://en.wikipedia.org/

[3] https://github.com/several27/FakeNewsCorpus

[4] https://www.kaggle.com/datasets/Cornell-University/arxiv

[5] http://mlg.ucd.ie/datasets/bbc.html

Datasets	FakeNews		20News		arXiv		NYT		BBCNews	
Metrics	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
					Text Classification					
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
cpt-text										
INSTRUCTOR			**80.62**	**78.72**	75.52	30.46	97.06	93.66	95.19	95.16
$\text{SIMDE}_{\text{bert}}$	**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
$\text{SIMDE}_{\text{long}}$	**71.60**	**61.66**	**75.44**	**74.38**	**77.68**	**33.26**	**97.90**	**95.43**	**96.67**	**95.91**
					Few-shot Text Classification					
BERT	23.96	23.73	19.94	18.71	24.08	10.14	51.85	43.90	54.22	52.73
CT-BERT	23.71	23.06	24.11	23.53	27.02	13.53	47.23	36.83	59.56	58.95
SimCSE	25.04	22.68	42.63	41.42	32.61	17.19	86.51	78.41	83.56	83.75
$\text{SimCSE}_{\text{long}}$	26.39	23.26	48.65	47.81	23.42	12.66	85.36	75.90	84.44	83.96
$\text{SIMDE}_{\text{bert}}$	**27.79**	**24.65**	**55.79**	**55.43**	**35.79**	**18.52**	**90.52**	**83.71**	**86.86**	**86.31**
LongFormer	26.56	25.12	44.42	42.41	25.04	13.36	73.06	54.87	84.89	85.47
BigBird	25.36	23.28	39.14	39.06	23.62	10.18	86.66	78.96	79.11	76.63
$\text{SIMDE}_{\text{long}}$	**29.17**	**27.13**	**51.18**	**50.96**	**34.33**	**18.80**	**89.78**	**82.88**	**86.78**	**86.66**

Table 2: For all performance measures, larger numbers are better. Our pre-trained model achieves the best results in all cases.

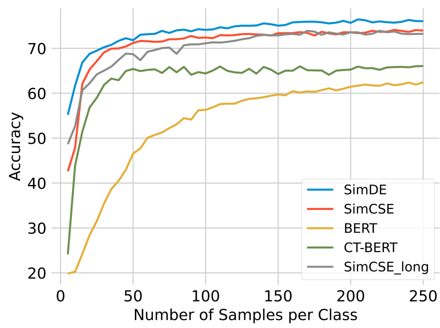

Figure 2: Performance of different models with different numbers of instances per class under few-shot setting.

the improvements from our method become larger as the number of training instances decreases indicating the importance of high-quality document embeddings for low-resource settings. Furthermore, our method achieves the best results under different numbers of training instances.

3.6 Document Retrieval

We use document retrieval to evaluate the ability of learning the similarity score between two vectors (**?**). 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 (**??**).

Results. Table **??** shows the results of document retrieval experiment on AAN dataset.

4 Conclusion

In this work, we propose an unsupervised contrastive learning framework for long text embeddings. Our method provides a new method for long text data augmentation without any supervision and language models can get large-scale pretraining on any long text. We conduct extensive experiments on text classification tasks under fully supervised and few-shot settings. Results show that our pretrained 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. Our method requires human-annotated data, which is expensive and time-consuming.
2. Our method focuses on document embeddings, we cannot claim the method works well for sentence embeddings.
3. Due to computing resource limitations, the text used in our work cannot be extremely long to verify the effectiveness of our method in this case.

Model	Retrieval
BERT	53.56
CT-BERT	54.23
SimCSE	54.78
SimCSE$_{\text{long}}$	55.23
cpt-text	
INSTRUCTOR	
SimDE$_{\text{bert}}$	58.29
LongFormer	56.89
BigBird	59.29
SimDE$_{\text{long}}$	59.89

Table 3

Ethics Statement

We do not anticipate any major ethical concerns; learning document embedding is a fundamental problem in natural language processing. Ethical considerations seem low-risk for the specific datasets studied here because they are all published.

A Appendix

A.1 Related Work

Text Embeddings. Learning text embeddings is a fundamental task in natural language processing (NLP) which can be used in information retrieval (**?**), text similarity (**?**), classification (**?**), etc. Existing works focus on learning general text embeddings for different applications. SimCSE (**?**) augment training data by Dropout (**?**) to construct positive data for contrastive learning. Sentence-BERT (**?**) employs siamese and triplet network structures to compare the labeled sentences in the natural language inference (NLI) dataset. Contriever (**?**) learns text embeddings for information retrieval via contrastive learning considering multilingual corpus. Instructor (**?**) collects a massive dataset from 330 diverse NLP tasks and formulates all tasks into information retrieval tasks with prompts. Previous works either rely on massive manually annotated data (**??**) or focus only on short text (i.e., sentences) embeddings (**?**). In this paper, we study unsupervised representation learning of document embeddings which usually have longer text than general sentence embeddings and this task brings new challenges for models to understand a document.

Model Structures for Documents. Due to the limitations of Transformer (**?**) on long documents, Another line of work studies the efficient model structures which can be effective to encode documents. For example, Longformer (**?**) introduces an attention mechanism that scales linearly with sequence length, making it easy to process long documents. BigBird (**?**) proposes a sparse attention mechanism that reduces this quadratic dependency to linear. **?** combines the memory mechanism in LSTM (**?**) with Transformer to let transformers encode extremely long sequences. Comparing these methods which focus on designing models for long documents, our work studies the effective training method for long documents which can be adopted to any model structures for pre-training.

A.2 Redundancy

We evaluate the redundancy of the text by two methods: (1) counting the repeated verbs and nouns in the text; (2) comparing inner-document and inter-document sentence similarities.

For (1), we first use SpaCy [6] to find verbs and nouns and get their lemmatizations. Intuitively, if the redundancy of a document is high, nouns and verbs will be repeated frequently to express the same topic. Hence, redundancies R in our paper are computed as:

$$R = \frac{N_{\text{nouns,verbs}}}{D_{\text{nouns,verbs}}} \tag{2}$$

where $N_{\text{nouns,verbs}}$ denotes the number of nouns and verbs in a document and $D_{\text{nouns,verbs}}$ is the number of distinct nouns and verbs.

For (2), we compute sentence similarities to compare inner-document and inter-document information. Specifically, we first split documents into sentences and encode these sentences with pre-trained SimCSE (**?**). Then, we calculate cosine similarities for every two sentences in the same documents (i.e., inner-document) and from different documents (i.e., inter-document). Results from five datasets are shown in Table **??**. We can see that sentences in the same documents have higher similarities compared to sentences from different documents. These results show that sentences in the same documents usually contain repeated information and hence have higher redundancy (i.e., sentences expressing similar semantics).

Length	(1)	(2)	(3)	(4)	(5)	All
FakeNews	1.06	1.21	29	1.35	1.52	1.37
20News	1.12	1.18	1.24	1.31	1.50	1.28
arXiv	1.12	1.25	1.36	1.49	1.62	1.34
NYT	1.00	1.14	1.21	1.31	1.48	1.45
BBCNews	1.05	1.14	1.20	1.29	1.46	1.19

Table 4: Information redundancies for different lengths (i.e., word numbers) of text: (1) 0-50 (2) 51-100 (3) 101-200 (4) 201-300 (5) more than 300.

Sentence Similarity	Inner-Document	Inter-Document
FakeNews	0.28	0.06
20News	0.21	0.07
arXiv	0.51	0.35
NYT	0.30	0.11
BBCNews	0.32	0.09

Table 5: Inner-document and inter-document sentences similarities measured by SimCSE.

A.3 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

[6] https://spacy.io/

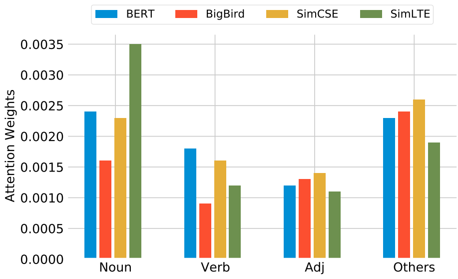

Figure 3: Attention weights from different models on the NYT dataset.

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.

A.4 Attention Weights

To explore the difference between SIMDE and other models, we analyze the attention weights of Transformers in different models on the NYT dataset (details in Appendix **??**). The average weights of different kinds of words are shown in Figure **??**. We can see that our model has more than 40% higher attention weights on nouns compared to BERT and SimCSE. **?** shows nouns are more informative than other words in the document understanding. Hence, our pretraining method increases the attention weights of models on nouns which results in higher performance on long text classification.

We compute the attention weights for Transformers as follows: (1) we first extract the attention weights between `[CLS]` token and all the other tokens; (2) we compute the averaged weights along different heads in multi-head attention; (3) the attention weights of the last layer in Transformers are used as the weights for words. Averaged values are computed for nouns, verbs, adjectives, and other words.