SimLTE: Simple Contrastive Learning for Long Text Embeddings

Anonymous ACL submission

Abstract

This paper presents SIMLTE, the first unsupervised pretraining method designed specifically for long text (e.g., documents, paragraphs). SIMLTE uses the contrastive learning framework, and our main contribution is a simple but effective data augmentation technique for generating similar text pairs. Specifically, we pretrain a language model to distinguish if two texts have the same topic without any supervision or specific model architectures, and so it is widely applicable. The positive pairs are constructed by our key information redundancy assumption for long text. On standard classification datasets, SIMLTE improves all baseline models, with an average improvement of 3.9% macro F1 score. We also consider a few-shot setting where we show an average improvement of 12.0%.

1 Introduction

Generating high quality text embeddings for long text is a long-standing open problem. Most previous studies focus on either sentence-level representations (Hill et al., 2016; Logeswaran and Lee, 2018; Gao et al., 2021) where training data usually contain short text or specific model structures allowing larger-range dependencies (Beltagy et al., 2020; Zaheer et al., 2020), but high-quality pretrained long text representations are less explored.

In this paper we present the SIMLTE which is the first unsupervised training method designed specifically for long text. The training procedure of SIMLTE can work with any model architecture to improve long text representations. Specifically, SIMLTE uses contrastive learning, and our key contribution is a new method for generating the positive samples for contrastive learning. To this end, we first investigate the information redundancy (details in Appendix A) on five datasets for different lengths of text. We find the information redundancy is larger as the length of the text is increasing. This

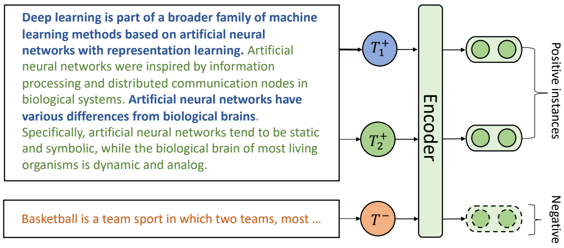

Figure 1: Overall framework of SIMLTE. The long text is randomly divided into two exclusive subsets of sentences and the two subsets work as positive pairs for contrastive learning. Other instances in the same batch are used as negatives.

result indicates long text usually contains repeated information. Based on this observation, we can assume that the model can still learn the main topic of a long text even if we drop some sentences. Hence, as shown in Figure 1, we randomly divide the original long text into two texts by sentences as positive pairs. Due to redundancies, our model can still recognize the two texts have the same topic. 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 text automatically.

To evaluate the quality of long text 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 SIMLTE 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 2 we formally define the contrastive learning problem and our novel SIMLTE training method. In Section 3 we develop a new experimental evaluation procedure for long text. We conclude in Section 4 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 SIMLTE 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 (Hadsell et al., 2006). 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 (Chen et al., 2020; Li et al., 2022) 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 (Devlin et al., 2019). 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 1.

2.2 SimLTE

The critical problem in contrastive learning is how to construct positive pairs (x, x^+). In representation learning for visual tasks (Chen et al., 2020), an effective solution is to take two random transformations of the same image (e.g., flipping, rotation). Similarly, in language representations, previous works (Gao et al., 2021; Karpukhin et al., 2020; Meng et al., 2021; Li et al., 2022) apply augmentation techniques such as dropout, word deletion, reordering, and masking.

In this paper, we propose a new method to construct positive instances for long text. 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 long text, we find long text (e.g., paragraphs) usually has higher information redundancy than short text (e.g., sentences) (Table 3 in Appendix). With this observation, we can have an assumption: the semantics of a long text will not be changed even if we drop half of the text. We can construct positive pairs under this assumption easily on any text dataset without supervision. Specifically, for each long text in the dataset, we randomly split sentences in the long text 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 long text to form two new texts. According to our assumption, the two new texts should have the same semantics and hence they are used as a positive pair in contrastive learning.

Consider an example (in Figure 1) to understand our positive instance construction process: Suppose we have a long text $T = (s_1, s_2, \ldots, s_n)$ where s_i is the i-th sentence in long text 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 long text 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 (Gao et al., 2021; Li et al., 2022).

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 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 SIMLTE.

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.

3.1 Pretraining Details

For pre-training, we start from the pretrained BERT-BASE model (Devlin et al., 2019) and the Longformer (Beltagy et al., 2020) model [1] We follow previous works (Gao et al., 2021; Li et al., 2022): 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 (Kingma and Ba, 2014) 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 (Gao et al., 2021).

3.2 Datasets

We use the following classic long text datasets to evaluate our method: (1) Fake News Corpus [3]; (2) 20NewsGroups (Lang, 1995); (3) arXiv articles dataset [4]; (4) New York Times Annotated Corpus (NYT) (Sandhaus, 2008); and (5) BBCNews [5]. We do not use semantic textual similarity (STS) tasks (Agirre et al., 2012) 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 (Devlin et al., 2019), SimCSE (Gao et al., 2021), CT-BERT (Carlsson et al., 2021). For a fair comparison, we also train a SimCSE with our pretraining dataset (SimCSE$_{\text{long}}$). (2) Transformers specified for long sequences include Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020). We train two versions of SimLTE with BERT and Longformer (i.e., SimLTE$_{\text{bert}}$ and SimLTE$_{\text{long}}$) for comparison. We do not include RoBERTa (Liu et al., 2019) and IS-BERT (Zhang et al., 2020) 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 B.

Results. Table 2 shows the evaluation results on different datasets. Overall, we can see that SimLTE achieves the best performance over the 5 long text datasets and consistently improves the long text 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 long text data (i.e., SimCSE$_{\text{long}}$) has similar results as the original SimCSE which indicates simplely increasing the length of pretraining text cannot improve long text 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 long text 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 B.

Results. Table 2 shows the results of few-shot text classification on these five datasets. We can see that SimLTE (i.e., SimLTE$_{\text{bert}}$ and SimLTE$_{\text{long}}$) 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 SimLTE$_{\text{bert}}$ with different numbers of training instances on 20News. The results in Figure 2 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
SimCSE$_{\text{long}}$	58.37	47.56	73.51	72.05	73.16	29.41	97.25	93.83	94.22	94.30
SimLTE$_{\text{bert}}$	**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
SimLTE$_{\text{long}}$	**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
SimCSE$_{\text{long}}$	26.39	23.26	48.65	47.81	23.42	12.66	85.36	75.90	84.44	83.96
SimLTE$_{\text{bert}}$	**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
SimLTE$_{\text{long}}$	**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.

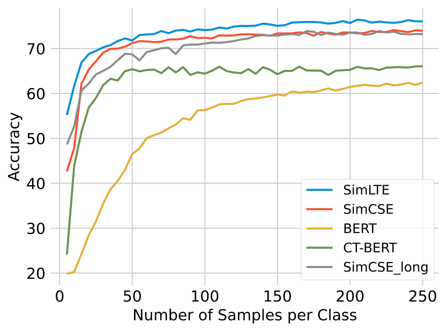

Figure 2: Performance of different models with different numbers of instances per class under few-shot setting.

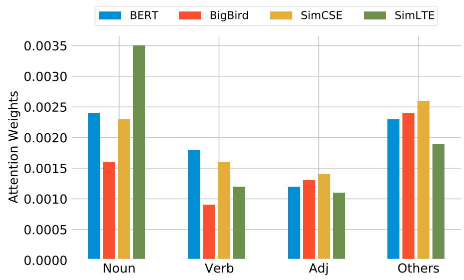

Figure 3: Attention weights from different models on the NYT dataset.

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

3.6 Attention Weights

To explore the difference between SimLTE and other models, we analyze the attention weights of Transformers in different models on the NYT dataset (details in Appendix C). The average weights of different kinds of words are shown in Figure 3. We can see that our model has more than 40% higher attention weights on nouns compared to BERT and SimCSE. Martin and Johnson (2015) 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.

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.

5 Ethical Concerns

We do not anticipate any major ethical concerns; learning text embeddings is a fundamental problem in natural language processing. We did not observe any such issues in our experiments, and indeed these considerations seem low-risk for the our datasets studied here because they are all published.

References

Eneko Agirre, Daniel Matthew Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEMEVAL*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *ICLR*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *NAACL*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *ICML*.

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. UCTopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6169, Dublin, Ireland. Association for Computational Linguistics.

Yinhan 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*, abs/1907.11692.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *ArXiv*, abs/1803.02893.

Fiona Martin and Mark Johnson. 2015. More efficient topic modelling through a noun only approach. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 111–115, Parramatta, Australia.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *ArXiv*, abs/2102.08473.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *ArXiv*, abs/2007.14062.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Conference on Empirical Methods in Natural Language Processing*.

A Redundancy

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 3: 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.

We evaluate the redundancy of the text by counting the repeated verbs and nouns in the text. Specifically, 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:

$$R = \frac{N_{\text{nouns,verbs}}}{D_{\text{nouns,verbs}}} \quad (2)$$

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.

B 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.

C Attention Weights

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.

[6] https://spacy.io/