Complex & Intelligent Systems (2023) 9:463–474
https://doi.org/10.1007/s40747-022-00819-1

ORIGINAL ARTICLE

Sentence part-enhanced BERT with respect to downstream tasks

Chaoming Liu[1] · Wenhao Zhu[1] · Xiaoyu Zhang[1] · Qiuhong Zhai[1]

Received: 14 December 2021 / Accepted: 24 June 2022 / Published online: 15 July 2022

Abstract
Bidirectional encoder representations from transformers (BERT) have achieved great success in many natural language processing tasks. However, BERT generally takes the embedding of the first token to represent sentence meaning in the tasks such as sentiment analysis and textual similarity, which does not properly treat different sentence parts. Different sentence parts have different levels of importance for different downstream tasks. For example, main parts (subject, predicate, and object) play crucial roles in textual similarity calculation, while secondary parts (adverbial and complement) are more important than the main parts in sentiment analysis. To this end, we propose a sentence part-enhanced BERT (SpeBERT) model that uses sentence parts with respect to downstream tasks to enhance sentence representations. Specifically, we encode sentence parts based on dependency parsing and downstream tasks, and extract embeddings through a pooling operation. Furthermore, we design several fusion strategies to incorporate different embeddings. We evaluate the proposed SpeBERT model on two downstream tasks, sentiment classification, and semantic textual similarity, with six benchmark datasets. The experimental results show that our model achieves better performance than competitor models.

Keywords Natural language processing · BERT · Sentence representation · Sentence part · Fusion strategy

Introduction

Developing effective models to represent the meaning of a sentence is a key task in the field of natural language processing (NLP). There are many applications of these representations of sentence meaning, such as text classification, textual similarity, paraphrase detection, and question answering.

In recent years, the pre-trained language model and its variants [1–4], such as BERT [2] have been widely utilized in text representations. BERT is a landmark pre-training model in the field of NLP, since it uses a transformer [5] to map the input into contextualized embeddings that are sensitive to the surrounding context, and obtains state-of-the-art results on a wide range of NLP tasks. BERT commonly applies the output of the first special token [CLS] as semantically meaningful representations. However, BERT ignores the influence of sentence parts on downstream tasks and does not consider the difference of sentence parts for downstream tasks accordingly.

A sentence consists of different parts such as subject, predicate, object, adverbial, and complement. Generally, different sentence parts have different levels of importance for different downstream tasks. As shown in Table 1, we observe that main parts (subject, predicate, and object) play more important roles than other parts (adverbial and complement) in the textual similarity task.[1] However, the other parts are more important than the main parts in the sentiment analysis task. Sentence-BERT (SBERT) [8] also shows that computing the mean of all the output vectors performs better than using only the embedding vector of the [CLS] token. Nevertheless, the same sentence parts play different levels of importance for different tasks. Instead of simply incorporating all sentence parts, we should further consider the effect of different sentence parts on downstream tasks.

✉ Wenhao Zhu
whzhu@shu.edu.cn

Chaoming Liu
cmliu488@shu.edu.cn

Xiaoyu Zhang
xiaoyu121@shu.edu.cn

Qiuhong Zhai
qiuhongzhai@gmail.com

[1] School of Computer Engineering and Science, Shanghai University, No. 99, Shangda Road, Shanghai 200444, China

[1] We train the base-scale model on the train set and then evaluate the model on the development set. We use accuracy as a metric for the STS-2 dataset while Spearman correlation is used for STS-B.

Table 1 Effect of sentence part on different downstream tasks

Sentence part	SST-2 [6]	STS-B [7]
Main parts	91.63	**88.46**
Other parts	**92.66**	87.82

The bold numbers represent the models which achieve best performance within same dataset

In this paper, we focus on utilizing sentence parts that are important to downstream tasks to enhance sentence representations, with the aim of introducing more helpful information. Consistent with the idea that different sentence parts have different levels of importance for different downstream tasks, we use sentence parts with respect to the downstream tasks to enhance sentence representations based on the BERT model. We name the proposed method SpeBERT. Previous models do not take advantage of the information from sentence parts, but this information is important for downstream tasks. In our model, we sufficiently consider useful information in terms of the sentence parts. More specifically, we first consider different sentence parts according to the differences in the downstream tasks. Then, we adopt a pooling operation to extract the embedding of sentence parts. Last, we use sentence representations enhanced by sentence parts in downstream tasks.

In the experiments, we show that the proposed model enhances sentence representations by incorporating important sentence parts. In contrast to previous work, our model uses sentence parts, which are important for downstream tasks, to enhance sentence representations. The experimental results of sentiment analysis and textual similarity calculation show that our proposed model significantly improves the model performance and achieves better results. In summary, our main contributions are listed as follows.

- We propose the idea of utilizing different sentence parts with respect to different downstream tasks to enhance the performance of sentence representations.
- We implement the idea of sentence parts enhancing on BERT. By using a pooling operation to extract the embedding of sentence parts, we design three strategies to merge different embeddings to render the incorporation more feasible.
- Our experiments on six datasets, including sentiment classification and semantic textual similarity tasks, show that SpeBERT achieves significant and consistent improvement in performance. Furthermore, our experiments on different downstream tasks show the importance of sentence parts that need to be considered for corresponding tasks, which may be useful in the research of downstream tasks.

Related work

In this section, we introduce BERT and then discuss some methods for learning semantically meaningful representations of sentences. BERT [2] obtains state-of-the-art results on eleven NLP tasks including natural language inference, named entity recognition, and question answering. BERT takes transformers [5] as backbone architecture. BERT is pre-trained on large-scale plain text for masked language modeling (MLM) and next sentence prediction (NSP) tasks. In the MLM task, 15% of the tokens in a sentence are processed in three ways. Specifically, 80%, 10%, and 10% of them are replaced by a MASK token, itself, or a random token, respectively. In the NSP task, two sentences A and B are stacked before feeding into BERT. Given 50% of the time when B is the next continuous sentence of A, BERT needs to leverage the representation of the [CLS] token to determine whether the input is continuous. The base version and large version of BERT have 12, and 24 transformer layers, respectively. When applying a pre-trained model to a specific downstream task, we only need to fine-tune the model with additional task-specific layers using the corresponding data.

Learning semantically meaningful representations of sentences is a well-studied area with many proposed methods. The simplest approach for generating sentence embeddings is to average word embeddings [9]. Skip-Thought vectors [10], which is an extension of the Skip-gram model for word embeddings [11] to sentences, learn reusable sentence representations from weakly labeled data. To address the problem that Skip-Thought vectors require weeks or months to train, Hill et al. [12] considered faster alternatives such as shallow log-linear models and sequential denoising autoencoders. Lin et al. [13] proposed a model for extracting an interpretable sentence embedding by introducing self-attention. They employed a 2D matrix to represent the embedding, with each row of the matrix attending to a different part of the sentence. Arora et al. [14] applied a weighted average of the word embeddings learned from an unlabeled corpus to represent the sentence and then transformed embeddings using the principal component analysis (PCA) or singular value decomposition (SVD) method.

Conneau et al. [15] showed that sentence representation learned from supervised data from the Stanford Natural Language Inference (SNLI) corpus [16] can consistently outperform unsupervised methods such as Skip-Thought. Yang et al. [17] proposed a method trained on conversations from Reddit using a Siamese transformer and Siamese deep averaging network (DAN) networks. Cer et al. [18] proposed a universal sentence encoder that trained encoders, including transformer and DAN network, and augmented unsupervised learning with training on the SNLI dataset.

Reimers and Gurevych [8] presented SBERT, a modification of the pre-trained BERT network that used Siamese and

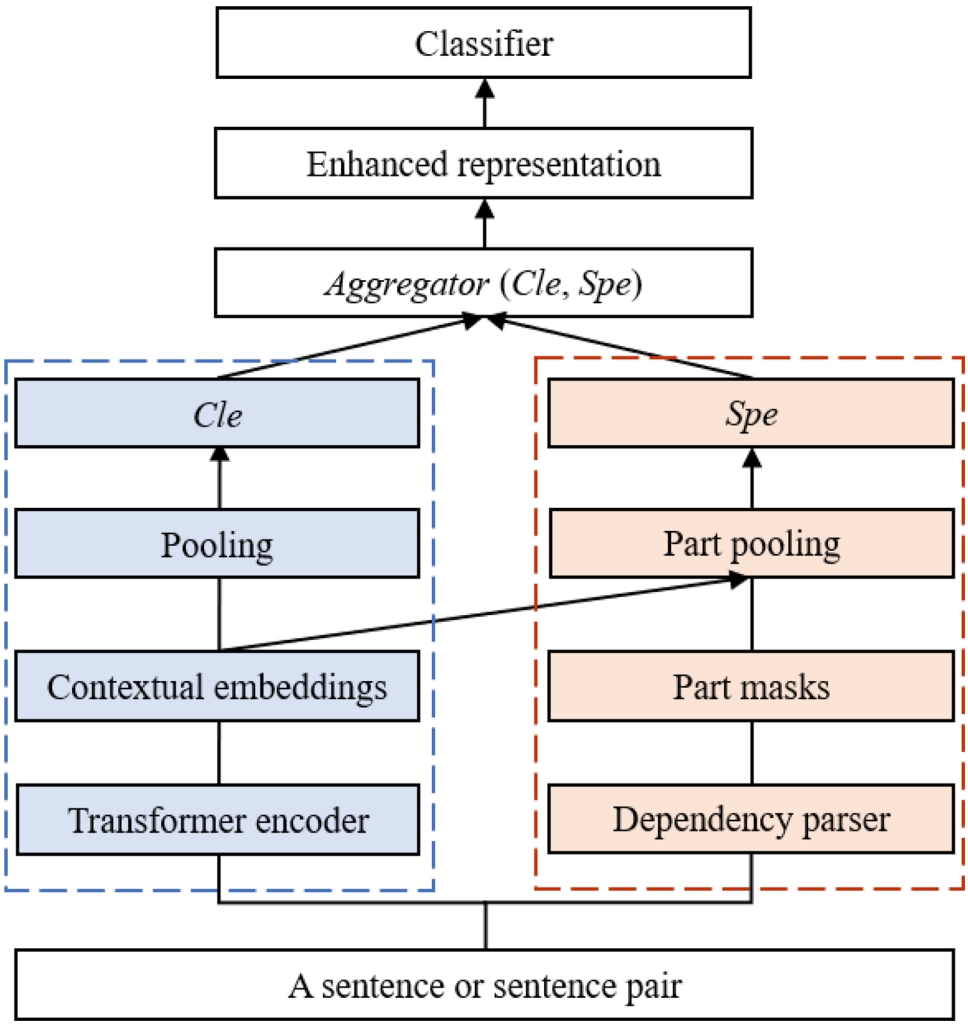

Fig. 1 An overview of SpeBERT. The blue dotted rectangle indicates the BERT applied to learn contextual embeddings and output the embedding of the [CLS] token. The red dotted rectangle indicates the component that is applied to extract the embedding of sentence parts. The sentence representations enriched by sentence parts are utilized for the downstream tasks

triplet network structures to derive semantically meaningful sentence embeddings that could be compared using cosine similarity. Ethayarajh [19] found that BERT word embeddings suffered from anisotropy, where the word embeddings occupied a narrow cone. Li et al. [20] argued that the semantic information in the BERT embeddings was not fully exploited while computing the similarity score using cosine similarity. More specifically, they proposed BERT-flow, which transformed the anisotropic sentence embedding distribution to a smooth and isotropic Gaussian distribution through normalizing flows that were learned with an unsupervised objective.

Using other useful information to help learn better language representations has also been explored. In word embedding learning, some works [21,22] tried to incorporate the phonetic, writing, and syntactic information into the text-based word representation models. In terms of sentence representation, Subramanian et al. [23] demonstrated that a multitask learning framework, which had many different training objectives in a single model, was beneficial for learning general-purpose sentence representations. Nie et al. [24] showed that dependency parsing and rule-based rubrics could curate a high-quality sentence relation task by leveraging explicit discourse relations.

Proposed method

In this section, we describe the SpeBERT model which leverages the sentence parts with respect to downstream tasks to enhance the performance of sentence representations. We present the main architecture and then introduce the details of some components of the model.

Overview

The architecture of our model is shown in Figure 1. Sentence representations are initially modeled by [CLS] embedding of BERT, and the representations are further enhanced via embedding of sentence parts. Two different pooling operations are applied to extract corresponding embeddings.

Enhanced sentence representations employed for the downstream task are generated by merging two embeddings via an aggregator. In particular, we aim to obtain representations that are enhanced by sentence parts, providing important information from sentence parts for the downstream task.

We briefly describe the BERT model, which usually takes a sentence or pairs of sentences as input. We denote this input simply as $S = \{s_1, s_2, \ldots, s_n\}$. Specifically, s_1 is always the [CLS] token. When S is packed by the sentence pair (S_1, S_2), the special token [SEP] is used to separate the two sentences. The input embeddings of BERT are formed by the sum of token embeddings, position embeddings, and segment embeddings. BERT uses a multilayer bidirectional transformer encoder to map the input embeddings (for simplification, we use X to denote the input embeddings of the sentence tokens) into context-aware embedding vectors. More specifically, the transformer encoder captures the contextual information of each word via self-attention and generates contextual embeddings:

$$H = Transformer\_encoder(X) \tag{1}$$

The sentence representation is produced by pooling the contextual embedding vectors:

$$C = Pooling(H) \tag{2}$$

The sentence representation is used to predict the target task:

$$O = softmax(W^T C) \tag{3}$$

where W is the trainable parameter matrix of the task.

In contrast to BERT, which only takes the embedding vector of the token [CLS] as the sentence representation, SpeBERT leverages different sentence parts based on the downstream tasks to further enhance the sentence representations. Concretely, we obtain the sentence part-enhanced representation by merging the [CLS] embedding with sentence part embedding via an aggregator. We discuss the extraction of different embeddings in Sect. 3.2. Currently, we simply use Cle and Spe to denote two kinds of embeddings. We apply fusion strategies, which are described in the Sect. 3.3, to merge two different embeddings and obtain the enhanced sentence representation:

$$C = Aggregator(Cle, Spe) \tag{4}$$

where *Aggregator* denotes the fusion methods. After obtaining the enhanced sentence representation, Eq. 3 is applied for prediction.

Extraction of different embeddings

By using the encoder in Eq. 1, we can obtain the contextual embeddings of the sentence. However, we need to further process the contextual embeddings to obtain the corresponding embeddings of the [CLS] token and sentence parts. We focus on the extraction of the sentence part embedding because the extraction of the [CLS] embedding is the same as that for the standard BERT. Therefore, we design a pooling operation, which is based on contextual embeddings and sentence part masks, to extract the embedding of the sentence parts. Next, we describe the sentence part masks and part pooling operation in detail.

Table 2 Example of dependency parsing

Sentence	Results of dependency parsing
The grain was terrible	((‘terrible’, ‘JJ’), ‘nsubj’, (‘grain’, ‘NN’))
	((‘grain’, ‘NN’), ‘det’, (‘The’, ‘DT’))
	((‘terrible’, ‘JJ’), ‘cop’, (‘was’, ‘VBD’))

Sentence part masks

To obtain the parts of a sentence, the Stanford dependency parser [25] is employed to yield dependency parsing. The dependency parser can output not only the dependency relations between words in the sentence but also the part of speech. Taking the sentence “The grain was terrible” as an example, the results are shown in Table 2. One triplet represents a dependency that occurs in the sentence. For example, in the triplet ((‘terrible’, ‘JJ’), ‘nsubj’, (‘grain’, ‘NN’)), the elements ‘JJ’ and ‘NN’ represent the part of speech of the corresponding words, and ‘nsubj’ denotes the dependency relationship between the words ‘terrible’ and ‘grain’.

We encode the sentence parts based on the results generated by the dependency parser. To distinguish the impact of different sentence parts on downstream tasks, we encode sentence parts with different values. More specifically, if a word belongs to the sentence parts that we take into account, the value of the corresponding mask is 1, and 0 otherwise.

Sentence part pooling

We leverage a pooling operation to derive the fixed-length embedding of the sentence parts. Unlike the pooling operation applied in standard BERT, part pooling is a design-specific mean pooling. Part pooling computes the sentence part embedding based on the masks of the sentence parts rather than directly averaging the embeddings of all sentence parts. The part pooling operation is represented as

$$Spe = \frac{1}{\sum m_i} \sum_i^n m_i \cdot h_i \tag{5}$$

where $m_i \in \{0, 1\}$ is the value at the i th element of sentence part masks and $h_i \in H$ is the embedding that corresponds to the i th sentence part.

Different fusion strategies

In our model, the enhanced sentence representations are obtained by fusing two different embeddings. To better merge the two embeddings, we design three kinds of fusion strategies and select the best strategy for the downstream task in our experiments.

Mean strategy

The mean strategy (MEAN) performs the elementwise mean operation between the [CLS] embedding and the part embedding.

$$C = Mean(Cle, Spe) \tag{6}$$

Concatenation strategy

The concatenation strategy (CONCAT) indicates that [CLS] embedding and part em-bedding are concatenated directly.

$$C = [Cle; Spe] \tag{7}$$

Weighted mean strategy

The weighted mean strategy (WMEAN) averages the two embeddings weighted by a learnable weight factor. The weight factor is inspired by the gate mechanism, which applies a neural network for computation.

$$C = (1 - \alpha) \cdot Spe + \alpha \cdot Cle \tag{8}$$

$$\alpha = \sigma(W^{'}[Cle; Spe] + b) \tag{9}$$

where [;] is the concatenation operator and $\sigma()$ is sigmoid function.

Model training

The sentence part-enhanced representation C is passed to a fully connected softmax classifier whose output is a probability distribution over the different classes. The model is trained through a backpropagation, where the objective function to be optimized is the cross-entropy loss defined as

$$J(\theta) = \sum_{s \in D} \sum_{n \in N} y_n(s)\ log\ \hat{y}_n(s) \tag{10}$$

where D is the collection of samples, N is the collection of distinct classes, and $y_n(s)$ is the ground truth for s. $\hat{y}_n(s)$ is the model's prediction for s. Note that in the semantical textual similarity task, s is constructed by concatenating two sentences, and $y_n(s)$ is the probability distribution transformed from the similarity score. θ represents trainable parameters for the model.

Experiment

We evaluate SpeBERT with sentiment analysis and textual similarity calculation. In particular, we choose six benchmark data sets, in which we need to consider different sentence parts for two different downstream tasks. We compare our model with the standard BERT and some competitor models. Furthermore, we conduct an ablation study to investigate the effect of the sentence part and fusion strategy in our proposed model.

Datasets

We use six freely available data sets, which cover two downstream tasks of sentiment classification and semantic textual similarity, as summarized in Table 3:

- MR: Sentiment prediction for movie reviews [26].
- CR: Sentiment prediction for customer product reviews [27].
- SST-2: Stanford Sentiment Treebank with binary labels [6].
- SICK-R: Semantic relatedness subtask from Sentence Involving Compositional Knowledge (SICK) [28].
- STS-B: Semantic Textual Similarity (STS) benchmark with sentence pairs derived from the three categories of captions, news, and forums [7].
- Chinese-STS-B: Chinese-STS-B is the Chinese version of STS-B, which is translated from the STS-B dataset [7] and further generates the rectified sentences by modifying wrong words and correcting some inaccurate sentences manually [29].

SICK-R and STS-B are two popular datasets used to evaluate STS tasks. The goal of STS tasks is to produce a score of how two semantically related sentences are based on the score annotated by humans. SICK-R and STS-B provide a value between 0 and 5 regarding the semantic relatedness of sentence pairs. SICK-R has a predefined split of 4,500 training pairs, 500 development pairs, and 4,927 testing pairs. STS-B includes 8,628 sentence pairs and is further divided into train (5,749), dev (1,500) and test (1,379). The Spearman's rank correlation is applied to evaluate the STS-B and Chinese-STS-B, while the Pearson correlation is used for SICK-R. The sentiment classification task considers classification accuracy as an evaluation metric.

Table 3 Summary of the two downstream tasks: sentiment classification and semantic textual similarity

Dataset	Task	#Number	#Label	Metrics
MR	Sentiment classification	11 k	2	Accuracy
CR	Sentiment classification	4 k	2	Accuracy
SST-2	Sentiment classification	70 k	2	Accuracy
SICK-R	Semantic similarity	10 k	1	Pearson correlation
STS-B	Semantic similarity	8.7 k	1	Spearman correlation
Chinese-STS-B	Semantic similarity	8.5 k	1	Spearman correlation

Experimental details

Throughout our experiment, we adopt the official TensorFlow code of BERT[2] as our codebase. In our experiments, we consider both uncased BERT-base and BERT-large for English datasets while Chinese-BERT-base is used for the Chinese-STS-B dataset. Note that our model is produced by fine-tuning the pre-trained BERT on a specific task dataset. For the finetuning of BERT, we follow the settings in [2]: Using Adam [30] as an optimizer with a learning rate of 2e-5, a batch size of 32, and a linear learning rate warm-up over 10% of the training data.

All the sentences are tokenized using WordPieces and are chopped to span a maximum length of 128 tokens. 10-fold cross-validation is performed to evaluate the MR and CR datasets, while the other datasets have a predefined split. With similarity score transformation [31], we treat the semantic textual similarity tasks as a classification problem. Our default fusion strategy is WMEAN for sentiment classification and CONCAT for semantic textual similarity tasks.

Experimental results

In Table 4 and Fig. 2, we report results that enrich BERT with sentence parts on two different downstream tasks. We also report results of sentiment classification under more evaluation metrics, as shown in Table 9. Our model achieves better results than BERT, which does not incorporate sentence parts with respect to downstream tasks. We attribute the obtained improvement to incorporating sentence parts, which indicates that SpeBERT can make good use of the semantic information introduced by the sentence parts. We will concretely discuss the results of two downstream tasks in the following sections.

Sentiment classification

We observe that the BERT-based models (including BERT, SBERT, and SpeBERT) achieve better results on the three datasets. InferSent and SBERT use the supervised natural language inference data to train a Siamese network and transfer learned representations. There, SBERT performs significantly better than InferSent. InferSent uses a Siamese bidirectional long short-term memory (Bi-LSTM) with max-pooling over the output, while SBERT uses a Siamese BERT with mean pooling. On the other hand, SBERT also performs the same trends compared to supervised methods trained from scratch on these datasets. More specifically, SBERT outperforms supervised models, such as Naive Bayes-SVM, on these datasets even if it is not fine-tuned on the corresponding training data. In addition, the impact of performance caused by the parameter size is obvious for the BERT-based models. For instance, the performance difference between BERT-base and BERT-large on the MR dataset is nearly 1.5% in terms of accuracy.

An interesting result is that BERT and SpeBERT outperform previous models with significant and consistent improvement. The difference between BERT and the other universal models is that BERT learns the sentence representations by fine-tuning directly on the target task, while other models aim to learn universal sentence representations and transfer them to the desired tasks. Taking SBERT as an example, it uses Siamese BERT networks and fine-tunes them on labeled natural language inference data to yield sentence representations that are used as features to train the classifier.

Unlike universal models, SpeBERT learns task-specific sentence representations using downstream information, similar to standard BERT and supervised models. However, we only finetune the pre-trained model on task-specific training data, which is different from previous supervised models trained from scratch. The task-specific sentence representations may illustrate why the performance of BERT is better than other models including the recent SBERT. Our proposed model further improves the performance since we use sentence parts to enhance sentence representations based on BERT.

Semantic textual similarity

We also evaluate the performance of SpeBERT on the STS task to determine how well sentence representations enhanced by sentence parts capture semantic similarity. As

[2] https://github.com/google-research/bert.

Table 4 Results of our model and competitor models on six benchmark datasets

Model	MR	CR	SST-2	SICK-R	STS-B	Chinese-STS-B
Unsupervised training methods						
FastSent [12]	70.80	78.40	–	–	–	–
FastSent+AE [12]	71.80	76.70	–	–	–	–
Skip-thought [10]	76.50	80.10	82.00	0.858	–	–
Skip-thought-LN [15]	79.40	83.10	82.90	0.858	70.20	–
Supervised training methods						
DictRep (bow) [15]	76.70	78.70	–	–	–	–
InferSent [15]	81.10	86.30	84.60	0.884	75.50	–
Multitask training methods						
LSMTL [23]	82.50	87.70	83.20	0.888	78.60	–
Self-supervised training methods						
DisSent books 5 [24]	80.20	85.40	82.80	0.845	–	–
DisSent books 8 [24]	79.80	85.00	83.90	0.854	–	–
DisSent books ALL [24]	80.10	84.90	84.10	0.849	–	–
Fine-tuned methods						
BERT-base[1]	87.44	90.69	93.25	0.884	84.34	79.94
BERT-large[1]	88.86	91.70	94.01	0.890	85.13	–
SBERT-base [8]	83.64	89.43	88.96	–	84.67	–
SBERT-large [8]	84.88	90.07	90.66	–	84.45	–
SpeBERT-base[1]	88.04	91.41	93.85	0.889	85.52	**80.86**
SpeBERT-large[1]	**89.47**	**92.65**	**94.84**	**0.894**	**86.37**	–
Supervised models trained from scratch (Results extracted from [23])						
Naive bayes-SVM	79.40	81.80	83.10	–	–	–
AdaSent	83.10	86.30	–	–	–	–
BLSTM-2DCNN	82.30	–	89.50	–	–	–

The bold numbers represent the models which achieve best performance within same dataset
[1]Models that we trained

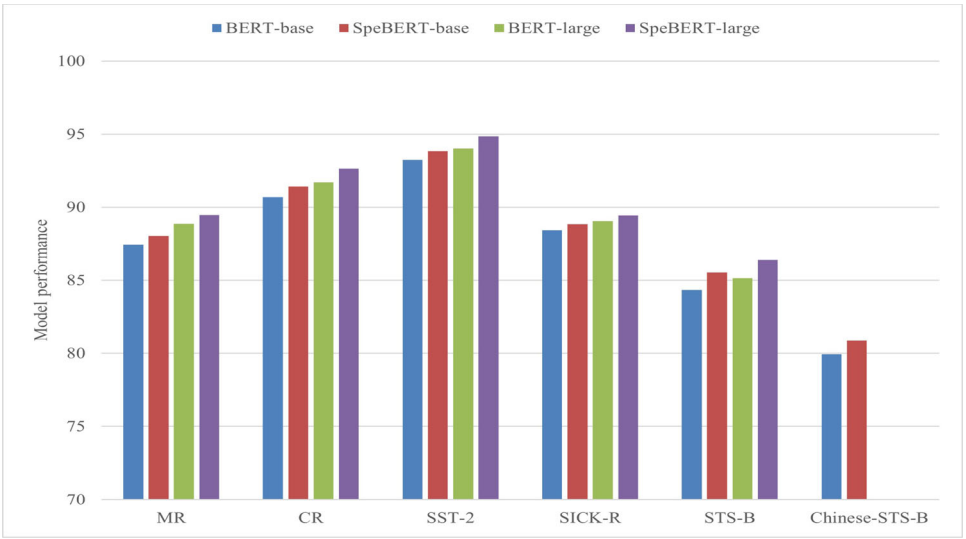

Fig. 2 The model performance of BERT and SpeBERT on six datasets

shown in Table 4, we observe that our model outperforms previous methods on two STS tasks. Specifically, our model achieves 0.4, 1.24, and 0.92 improvements in performance compared with BERT on the SICK-R dataset, STS-B dataset, and Chinese-STS-B dataset, respectively. This indicates that incorporating the sentence parts is advantageous for the textual similarity task.

Although all the models learn sentence representations via different methods, we note that the performance improvement of the SICK-R dataset is not always obvious. However, our model outperforms other models and achieves the best results on this dataset. Moreover, our model is comparable to the large-scale multitask learning (LSMTL) model on the SICK-R dataset, where the representations are learned on five language-related tasks. This indicates that SpeBERT is capable of capturing semantic information, even though it is only trained on the target task.

Compared to SBERT, our model also achieves better performance. We find that our model, similar to BERT and SBERT, obtains an obvious gain in performance compared with the other models. One possible explanation is that the pre-trained model has learned useful knowledge from a vast amount of text during the pre-training stage, and further captures useful semantic information through fine-tuning on downstream tasks. Moreover, SpeBERT-large performs better than SpeBERT-base. This is in contrast to [8], who found it beneficial for SBERT with the base model instead of the large model.

Ablation study

We have demonstrated the experimental results of SpeBERT on sentiment classification and semantic textual similarity. In this section, we conduct an ablation study on the different aspects of SpeBERT to better understand their relative importance.

In our ablation study, we use the SST-2 and STS-B datasets to better investigate the impact of the sentence part and the fusion strategy on downstream tasks. We fine-tune the base model on the corresponding training set, and performances are measured on the development set. The results are shown in Table 5.

Table 5 Impact of the sentence part and fusion strategy on model performance

Ablation setting	SST-2	STS-B
Sentence part		
w/ main parts	92.20	**89.01**
w/ other parts	**92.78**	88.37
w/o (BERT)	92.55	88.56
Fusion strategy		
MEAN	92.78	89.01
CONCAT	92.66	**89.30**
WMEAN	**93.12**	88.89

The bold number represent the model which achieves the best results on same dataset within specific ablation setting

Sentence part

We evaluate the importance of different sentence parts on different downstream tasks. Specifically, we demonstrate the effect of different sentence parts on sentiment analysis and textual similarity calculation tasks when using the MEAN strategy as a fusion strategy. We find that models with different sentence parts do not always achieve the best result in both tasks. For example, SpeBERT with main parts (subject, predicate, and object) outperforms SpeBERT with other parts (adverbial and complement) on textual similarity calculation while underperforming on sentiment analysis. This finding indicates that we need to emphasize different sentence parts according to different downstream tasks, rather than incorporating the same sentence parts for different tasks. In addition, the model with main parts underperforms BERT on the SST-2 dataset while incorporating other parts also achieves similar results on STS-B. Therefore, incorporating inappropriate sentence parts may adversely affect model performance and even yield performance degradation. As a result, we should strengthen the sentence parts with respect to downstream tasks and minimize the influence of noise when we try to leverage more sentence information.

Fusion strategy

As shown in Fig. 3, we evaluate the different fusion strategies (MEAN, CONCAT and WMEAN) and observe that the impact of the fusion method is obvious. Specifically, SpeBERT with different fusion strategies outperforms BERT on the SST-2 and STS-B datasets. However, different fusion strategies lead to different effects on different downstream tasks. For example, SpeBERT with the WMEAN strategy achieves the best results on the SST-2 dataset, while SpeBERT with the CONCAT strategy obtains the best results on the STS-B dataset. This indicates that the same strategy does not always achieve the best results on different downstream tasks. As a result, we should adopt a task-dependent strategy to adequately fuse different embeddings. Additionally, SpeBERT with the MEAN strategy, which simply averages the two embeddings, outperforms BERT on two different downstream tasks. This result shows that we can use a simple and promising method to improve BERT-based models without modifying the original models.

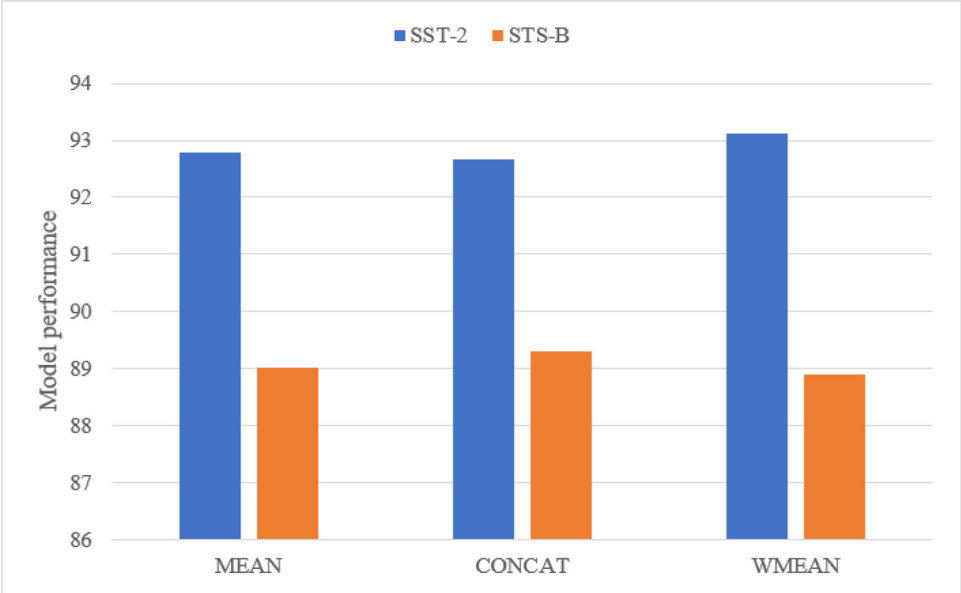

Fig. 3 Effect of different fusion strategies on the model performance

Table 6 Instances of ground truth and model predictions

Sentence	Ground truth	BERT	SpeBERT
Is n't it great ?	1	0	1
A very average science fiction film.	0	1	0
Propelled not by characters but by caricatures.	0	1	0

Analysis

Case analysis

To better analyze the model capacity of the original BERT and our proposed SpeBERT, we take some instances from SST-2 dataset to discuss the difference between ground truth and model predictions.

As shown in Table 6, we observe that SpeBERT usually outperforms the BERT when a review sentence has a complex expression. Specifically, SpeBERT further enhances the sentence representations by incorporating sentence parts with respect to tasks based on BERT, in which sentence parts capture the useful information for sentiment classification. This may explain that SpeBERT can correctly classify while BERT cannot for some hard examples. This indicates that sentence part enhanced SpeBERT not only has better performance but also has better robustness compared to the original BERT.

Parameter size

Generally, the model size is an important factor in model performance. As shown in Table 4, large models usually perform better than base models. To compare the difference in model parameters, we count the model size on different downstream tasks and show it in Table 7.

From Table 7, we observe that the parameter size of large models is about three times larger than the base model. Our SpeBERT has more parameters since it incorporates the sentence parts to enhance sentence representations. Compared to standard BERT, the external parameters of our proposed model are brought by fusion strategy. We take the model size of SpeBERT-base on MR dataset as an example, in which we use weighted mean strategy (WMEAN) to fuse the two embeddings and finally obtain the enhanced sentence representations, the parameter number of fusion strategy is computed by: 768*2+1=1537. This means that the parameter difference between SpeBERT-base and BERT-base is 1537 (from 110074370 to 110075907). In summary, our proposed SpeBERT has similar model complexity to BERT while our model further improves the model performance with limited parameter growth.

Model runtime

To evaluate the model's computational efficiency, we report the model runtime on different datasets. Performances are measured on Nvidia Tesla V100 GPU, CUDA 10.2 and cuDNN. The results are depicted in Table 8.

We observe that most models have low runtime with the help of GPU. The base models are almost three times faster than large models, and the model usually spends more time

Table 7 Parameter size of different models

Model	MR	CR	SST-2	SICK-R	STS-B	Chinese-STS-B
BERT-base	110074370	110074370	110074370	110076677	110077446	102862854
BERT-large	336193538	336193538	336193538	336196613	336197638	–
SpeBERT-base	110075907	110075907	110075907	110080517	110082054	102867462
SpeBERT-large	336195587	336195587	336195587	336201733	336203782	–

Table 8 Model runtime of BERT and SpeBERT

Model	MR	CR	SST-2	SICK-R	STS-B	Chinese-STS-B
BERT-base	328s	143s	2022s	189s	214s	198s
BERT-large	943s	392s	6070s	534s	609s	–
SpeBERT-base	327s	142s	2028s	190s	216s	199s
SpeBERT-large	947s	392s	6073s	531s	610s	–

on a large dataset. Compared to BERT, SpeBERT has a competitive time cost and even has less model runtime. This may be due to our SpeBERT enhanced by the sentence parts with respect to downstream tasks speeds up the convergence of the model. However, BERT which does not utilize sentence parts only views the embedding of the first special token as sentence representation.

Conclusion

In this work, we present SpeBERT, a sentence part-enhanced BERT model that utilizes sentence parts with respect to downstream tasks to enhance sentence representations. We incorporate different sentence parts based on the differences in downstream tasks and design a pooling operation to extract the embedding of the sentence parts. We also adopt three kinds of strategies to sufficiently fuse two different embeddings. The experimental results demonstrate that in the context of sentiment classification and semantic textual similarity tasks, our model performs better than competitor models and standard BERT. Furthermore, we show the importance of sentence parts that need to be considered for corresponding tasks, which may be helpful during the research of downstream tasks.

There are many future areas to explore to improve SpeBERT, including more novel pre-trained models [32,33], ways of combining the different embeddings in a more complex manner, and a more effective method that weights each sentence part, such as using learned external parameters to weight the different sentence parts independently and using negations to update the weights of the sentence parts that are unimportant for downstream tasks. At last, we also would like to verify whether SpeBERT is effective against other more downstream tasks.

Acknowledgements This research was funded by the National Natural Science Foundation of China (61572434), and the Shanghai Science and Technology Committee (19DZ2204800).

Author Contributions CL: Conceptualization, methodology, writing–original draft. WZ: Writing–reviewing and editing, supervision. XZ: Writing–reviewing and editing, validation. QZ: Data curation, validation.

Data Availability The datasets employed in this paper are publicly available.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Code availability As the current research is still in progress, we decided not to share the code for the time being.

Appendix A: Experimental results under more evaluation metrics

See Table 9 here.

Table 9 Experimental results of sentiment classification under more evaluation metrics

Dataset	Metrics	BERT-base	BERT-large	SpeBERT-base	SpeBERT-large
MR	Precision	87.00	88.06	87.67	89.37
	Recall	88.04	89.86	88.52	89.57
	F1 score	87.49	88.95	88.08	89.47
	Accuracy	87.44	88.86	88.04	89.47
CR	Precision	91.94	92.71	92.00	92.69
	Recall	93.79	94.39	94.78	96.08
	F1 score	92.77	93.53	93.36	94.34
	Accuracy	90.69	91.70	91.41	92.65
SST-2	Precision	90.77	91.24	91.82	93.03
	Recall	96.26	97.36	97.26	96.92
	F1 score	93.43	94.20	93.98	94.94
	Accuracy	93.25	94.01	93.85	94.84

References

1. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf
2. Devlin J, Chang M-W, Lee K, Toutanova KN (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol 1 (Long and Short Papers), pp 4171–4186
3. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol 32
4. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, vol 30, pp 5998–6008
6. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp 1631–1642
7. Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L (2017) Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, Canada, pp 1–14
8. Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 3980–3990
9. Kusner M, Sun Y, Kolkin N, Weinberger K (2015) From word embeddings to document distances. In: Proceedings of The 32nd International Conference on Machine Learning, pp 957–966
10. Kiros R, Zhu Y, Salakhutdinov R, Zemel RS, Torralba A, Urtasun R, Fidler S (2015) Skip-thought vectors. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol 28, pp 3294–3302
11. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, vol 26, pp 3111–3119
12. Hill F, Cho K, Korhonen A (2016) Learning distributed representations of sentences from unlabelled data. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1367–1377
13. Lin Z, Feng M, dos Santos CN, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. In: ICLR 2017 : International Conference on Learning Representations 2017
14. Arora S, Liang Y, Ma T (2017) A simple but tough-to-beat baseline for sentence embeddings. In: ICLR 2017 : International Conference on Learning Representations 2017
15. Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A (2017) Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp 670–680
16. Bowman SR, Angeli G, Potts C, Manning CD (2015) A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp 632–642
17. Yang Y, Yuan S, Cer D, Kong S-y, Constant N, Pilar P, Ge H, Sung Y-H, Strope B, Kurzweil R (2018) Learning semantic textual similarity from conversations. In: Proceedings of The Third Workshop on Representation Learning for NLP, Association for Computational Linguistics, Melbourne, Australia, pp 164–174
18. Cer D, Yang Y, Kong S-y, Hua N, Limtiaco N, John RS, Constant N, Guajardo-Cespedes M, Yuan S, Tar C, Sung Y-H, Strope B, Kurzweil R (2018) Universal sentence encoder. arXiv preprint arXiv:1803.11175
19. Ethayarajh K (2019) How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 55–65
20. Li B, Zhou H, He J, Wang M, Yang Y, Li L (2020) On the sentence embeddings from pre-trained language models. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online

21. Zhu W, Jin X, Liu S, Lu Z, Zhang W, Yan K, Wei B (2020) Enhanced double-carrier word embedding via phonetics and writing. ACM Transactions on Asian and Low-Resource Language Information Processing 19(2)
22. Zhu W, Liu S, Liu C (2021) Learning multimodal word representation with graph convolutional networks. Information Processing & Management 58(6):102709
23. Subramanian S, Trischler A, Bengio Y, Pal CJ (2018) Learning general purpose distributed sentence representations via large scale multi-task learning. In: International Conference on Learning Representations
24. Nie A, Bennett E, Goodman ND (2019) Dissent: Learning sentence representations from explicit discourse relations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 4497–4510
25. Chen D, Manning C (2014) A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp 740–750
26. Pang B, Lee L (2005) Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp 115–124
27. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 168–177
28. Marelli M, Menini S, Baroni M, Bentivogli L, bernardi R, Zamparelli R (2014) A sick cure for the evaluation of compositional distributional semantic models. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pp 216–223
29. Zeng J (2020) A large-scale chinese nature language inference and semantic similarity calculation dataset. https://6a75-junzeng-uxxxm-1300734931.tcb.qcloud.la/CNSD.pdf
30. Kingma DP, Ba JL (2015) Adam: A method for stochastic optimization. In: International Conference on Learning Representations
31. Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, vol 1 (Long Papers), pp 1556–1566
32. Yan Y, Li R, Wang S, Zhang F, Wu W, Xu W (2021) ConSERT: A contrastive framework for self-supervised sentence representation transfer. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, pp 5065–5075
33. Gao T, Yao X, Chen D (2021) SimCSE: Simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp 6894–6910