Comparison and Combination of Sentence Embeddings Derived from Different Supervision Signals

Hayato Tsukagoshi **Ryohei Sasano** **Koichi Takeda**
Graduate School of Informatics, Nagoya University
tsukagoshi.hayato.r2@s.mail.nagoya-u.ac.jp,
{sasano,takedasu}@i.nagoya-u.ac.jp

Abstract

There have been many successful applications of sentence embedding methods. However, it has not been well understood what properties are captured in the resulting sentence embeddings depending on the supervision signals. In this paper, we focus on two types of sentence embedding methods with similar architectures and tasks: one fine-tunes pre-trained language models on the natural language inference task, and the other fine-tunes pre-trained language models on word prediction task from its definition sentence, and investigate their properties. Specifically, we compare their performances on semantic textual similarity (STS) tasks using STS datasets partitioned from two perspectives: 1) sentence source and 2) superficial similarity of the sentence pairs, and compare their performances on the downstream and probing tasks. Furthermore, we attempt to combine the two methods and demonstrate that combining the two methods yields substantially better performance than the respective methods on unsupervised STS tasks and downstream tasks.

1 Introduction

Sentence embeddings are dense vector representations of a sentence. A variety of methods have been proposed to derive sentence embeddings, including those based on unsupervised learning (Kiros et al., 2015; Hill et al., 2016; Logeswaran and Lee, 2018; Cer et al., 2018; Wang et al., 2021) and supervised learning (Conneau et al., 2017). Pre-trained Transformer-based (Vaswani et al., 2017) language models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), have been successfully applied in a wide range of NLP tasks, and sentence embedding methods that leverage pre-trained language models have also performed well on semantic textual similarity (STS) tasks and several downstream tasks. These methods refine pre-trained language models for sophisticated sentence embeddings by unsupervised learning (Li et al., 2020; Wang and Kuo, 2020; Giorgi et al., 2021; Carlsson et al., 2021; Yan et al., 2021; Gao et al., 2021), or supervised learning (Reimers and Gurevych, 2019; Tsukagoshi et al., 2021; Gao et al., 2021).

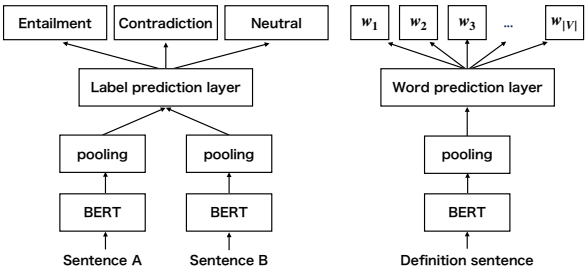

Figure 1: Overviews of SBERT (left) and DefSent (right).

Among them, Reimers and Gurevych (2019) proposed Sentence-BERT (SBERT), which fine-tunes pre-trained language models on the natural language inference (NLI) task. SBERT performed well on the STS and downstream tasks. Recently, Tsukagoshi et al. (2021) proposed DefSent, which fine-tunes pre-trained language models on the task of predicting a word from its definition sentence in a dictionary, and reported that it performed comparably to SBERT. Figure 1 shows overviews of SBERT and DefSent. Although both methods fine-tune the same pre-trained models and use the same pooling operations to derive a sentence embedding, the supervision signals for fine-tuning are different. That is, SBERT leverages NLI datasets, whereas DefSent leverages word dictionaries.

It is expected that the properties of the sentence embeddings depend on their supervision signals. However, since existing research has mainly focused on achieving better performance on benchmark tasks, it has not been revealed what property differences the resulting sentence embeddings have. Investigating the properties of sentence embeddings would give us a better understanding of existing sentence embedding methods and help develop further methods. In this paper, we empirically investigate the influence of supervision signals on

Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, pages 139 - 150

sentence embeddings. We focus on SBERT and DefSent because they leverage different supervision signals but have very similar architectures, as shown in Figure 1; thus, they would be appropriate for analyzing the influence of the supervision signals on sentence embeddings.

First, we partitioned the STS datasets (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017; Marelli et al., 2014) on the basis of two different perspectives and examine what type of meaning each type of sentence embeddings captures by analyzing the performance of each method on these partitioned STS datasets. We then apply each type of embeddings to the downstream and probing tasks of SentEval (Conneau and Kiela, 2018) and analyze what type of information is captured. Our results demonstrate that the supervision signals have a significant impact on performance on these tasks and that the properties of SBERT and DefSent would be complementary. Thus, we further explore whether combining the two methods yields better sentence embeddings to confirm their complementarity, and demonstrate that combining the two methods yields substantially better performance than the respective methods on unsupervised STS tasks and downstream tasks of SentEval.

2 Preparation

In this section, we present detailed descriptions of SBERT and DefSent, the two sentence embedding methods compared in this study, and describe the tasks and settings for the experiments.

2.1 Sentence-BERT

Sentence-BERT (SBERT) proposed by Reimers and Gurevych (2019) is a sentence embedding method that fine-tunes pre-trained language models in a Siamese network architecture on the NLI task. An overview of SBERT is given on the left side of Figure 1[1]. For fine-tuning of SBERT, NLI datasets, such as the Stanford NLI (SNLI) dataset (Bowman et al., 2015) and Multi-Genre NLI (MultiNLI) dataset (Williams et al., 2018), are used. These datasets consist of sentence pairs labeled as either entailment, contradiction, or neutral. The NLI task is a classification task to predict these labels.

SBERT first inputs each sentence of a pair into BERT and obtains sentence embeddings from the output contextualized word embeddings by a pooling operation. SBERT uses three types of pooling strategies: `CLS`, which uses the embedding of the first token of the input sequence (e.g., the [CLS] token for BERT); `Mean`, which uses the average of all word embeddings; and `Max`, which uses the max-over-time of all word embeddings. Let u and v be the sentence embeddings obtained by such pooling. SBERT composes a vector $[u; v; |u - v|]$ and inputs it into a three-way softmax classifier to predict the label of the given sentence pair.

2.2 DefSent

DefSent proposed by Tsukagoshi et al. (2021) is a sentence embedding method that fine-tunes pre-trained language models on the task of predicting a word from its definition sentence in a dictionary. An overview of DefSent is given on the right side of Figure 1. As well as SBERT, DefSent first inputs a definition sentence into BERT and obtains the sentence embedding by a pooling operation, which uses `CLS`, `Mean`, and `Max` as the pooling strategies. The derived sentence embedding is then input to the word prediction layer and fine-tunes the model to predict the corresponding word. The word prediction layer is the one that was used for masked language modeling during pre-training. Tsukagoshi et al. (2021) reported that DefSent performed comparably to SBERT.

2.3 STS tasks

We use STS tasks to investigate the properties of sentence embeddings. STS tasks evaluate how the semantic similarity between two sentences calculated with a model correlates with a human-labeled similarity score through Pearson and Spearman correlations. There are two types of settings: supervised and unsupervised. In the supervised setting, a model learns a regression function that maps a pair of sentences to a similarity score using some of the STS datasets. In the unsupervised setting, no training is performed on STS datasets, and we compute the similarity between two sentence embeddings, with a similarity score such as cosine similarity.

For the evaluation of the STS tasks, STS12–STS16 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017), and SICK-R (Marelli et al., 2014) are often used. Each dataset contains sentence pairs with their semantic similarity scores as gold labels given by real numbers ranging from 0 to 5. Each of the STS12–STS16 datasets consists of sentence pairs from multiple sources. For example, STS12 consists of sen-

[1] Actually, it is possible to use RoBERTa and others instead of BERT, but for simplicity we refer to it as BERT here.

Sources		#	Origin
STS12	*MSRpar*	750	newswire
	MSRvid	750	videos
	SMTeuroparl	459	WMT eval.
	OnWN	750	glosses
	SMTnews	399	WMT eval.
STS13	*FNWN*	189	glosses
	headlines	750	newswire
	OnWN	561	glosses
STS14	*deft-forum*	450	forum posts
	deft-news	300	news summary
	headlines	750	newswire headlines
	images	750	image descriptions
	OnWN	750	glosses
	tweet-news	750	tweet-news pairs
STS15	*answers-forums*	375	Q&A forum answers
	answers-students	750	student answers
	belief	375	committed belief
	headlines	750	newswire headlines
	images	750	image descriptions
STS16	*answer-answer*	254	Q&A forum answers
	headlines	249	newswire headlines
	plagiarism	230	short-answer plag.
	postediting	244	MT postedits
	question-question	209	Q&A forum questions

Table 1: Statistics of STS datasets partitioned by source. "#" denotes number of sentence pairs, and "Origin" denotes origin of dataset.

tence pairs from five sources: *MSRpar*, *MSRvid*, *SMTeuroparl*, *OnWN*, and *SMTnews*. Table 1 lists the sources of each dataset in STS12–STS16.

2.4 SentEval

We also compare SBERT and DefSent on SentEval (Conneau and Kiela, 2018) tasks. SentEval is a widely used toolkit to evaluate the quality of sentence embeddings by measuring the performance on classification tasks. Since SentEval provides various classification tasks, it is suitable for investigating the properties of sentence embeddings. SentEval consists of two types of tasks: downstream tasks and probing tasks. Downstream tasks are binary or multi-class classification tasks, such as sentiment classification in movie reviews and question-type classification. Probing tasks are classification tasks for linguistic information, such as sentence length and tense classification.

2.5 Experimental settings

In the experiments reported in Sections 3 and 4, we use BERT-base (bert-base-uncased), BERT-large (bert-large-uncased), RoBERTa-base (roberta-base), and RoBERTa-large (roberta-large) from Transformers (Wolf et al., 2020) as the pre-trained language models and adopt `Mean` as the pooling strategy. We use the same settings as Reimers and Gurevych (2019) and Tsukagoshi et al. (2021) for fine-tuning. We provide further training details in Appendix A, and report the fine-tuning time and computing infrastructure in Appendix B.

3 Comparison of Sentence Embeddings

The supervision signal used for fine-tuning sentence embeddings might affect their properties. For example, since it is crucial to capture the differences in meaning even when the given sentence pair is superficially similar in the NLI task, SBERT is considered suitable for determining the semantic similarity between superficially similar sentence pairs. In this section, we attempt to reveal such properties of each type of sentence embeddings. First, we partition the STS datasets on the basis of the source of the sentence pairs and the superficial similarity of the sentence pair. We then apply each type of embeddings to the downstream and probing tasks of SentEval.

3.1 STS partitioned by source

We assume that each sentence embedding method might better capture the meaning of sentences similar to those in the dataset used for fine-tuning, i.e., NLI datasets for SBERT and word dictionaries for DefSent. Thus, we partition STS12–STS16 datasets in accordance with the source of the sentences and measure the performance for each subset. We adopt the unsupervised setting. We calculate Spearman's rank correlation coefficient (ρ) between semantic similarity scores and each type of sentence embeddings. For comparison, we conduct evaluations on the concatenation of all subsets, i.e., the STS datasets without partitioning. We fine-tune and evaluate SBERT and DefSent 10 times with different seed values and report the average. We also evaluate the model without fine-tuning (w/o FT) for comparison.

Figure 2 shows the Spearman's ρ for the subsets of the STS12–STS16 datasets. It is worth noting that since we use correlations, the evaluation score on the concatenation of all subsets is not the average of the other scores, and in extreme cases it can be smaller than the minimum of the other scores. We can see that both SBERT and DefSent achieve higher scores than w/oFT on most subsets. Although DefSent consistently performs better than w/oFT in all subsets, SBERT performs worse than w/oFT in some subsets. Comparing SBERT and DefSent, when we focus on individual subsets, we can find that there are cases in

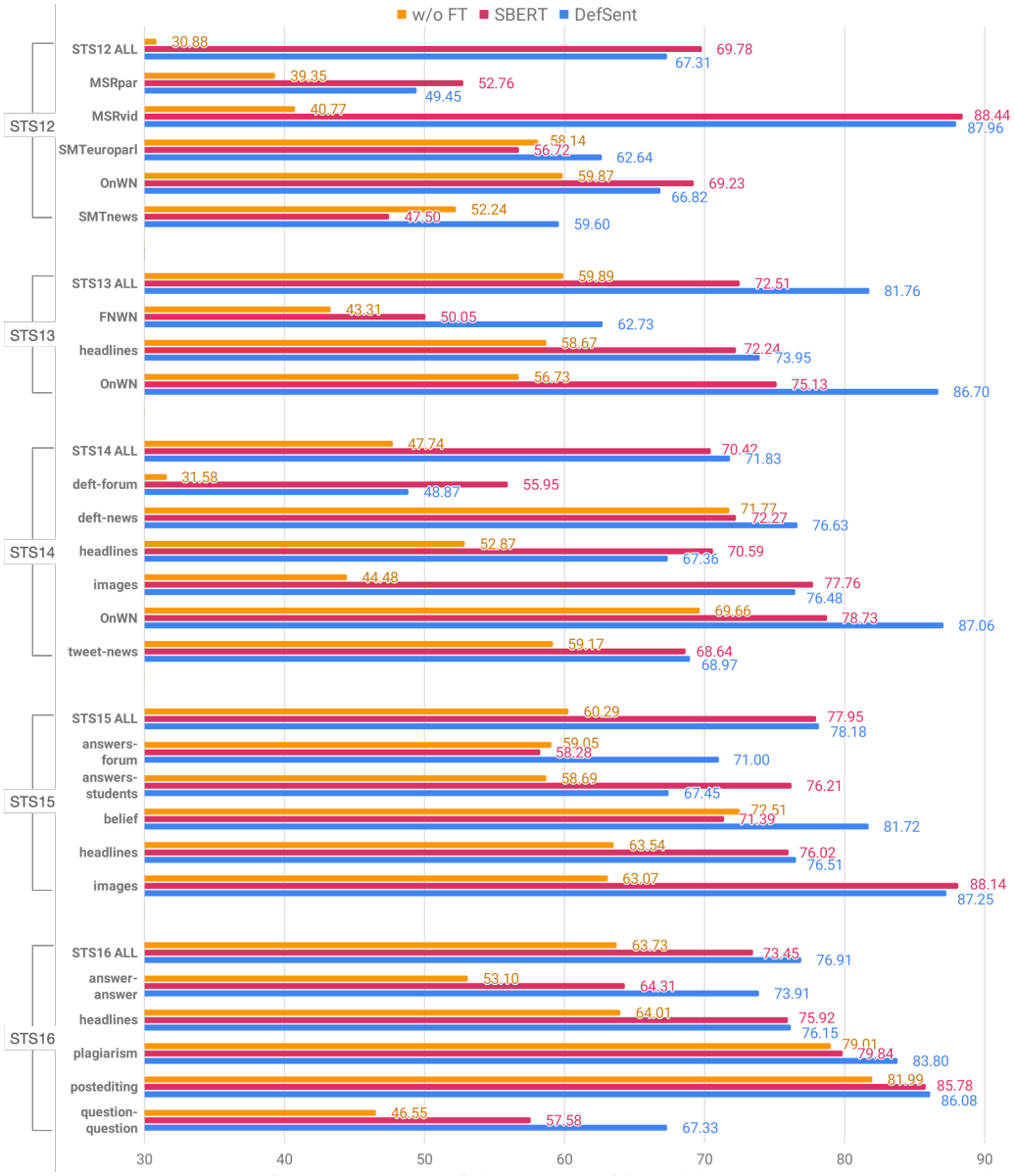

Figure 2: Spearman's $\rho \times 100$ for STS12–STS16 datasets partitioned by source. "STS# ALL" denotes the concatenation of all subsets for each STS dataset.

which SBERT achieves higher scores than DefSent, but we can say that DefSent achieves slightly higher scores as a whole. DefSent achieves noticeably higher scores than SBERT on *OnWN* and *FNWN* of STS13 and *OnWN* of STS14. *OnWN* and *FNWN* of STS13 are datasets created using definition sentences in OntoNotes, FrameNet, and WordNet. These results, as expected, indicate that DefSent is capable of adequately representing the meaning of definition sentences. However, SBERT achieves higher scores than DefSent on *deft-forum* and *headlines* of STS14 and *answer-students* of STS15. Regarding *answer-students*, since it is built from a dataset that has a similar format to the NLI datasets (Agirre et al., 2015), it is considered a score such as the one observed is as expected for SBERT, which is trained on the NLI datasets.

3.2 STS partitioned by Dice coefficient

We then explore how the similarity of sentence embeddings is affected by the superficial similarity of the sentences. Generally speaking, it is considered difficult to correctly order the similarity of a dataset consisting of pairs with high superficial similarity. However, since the NLI datasets contain a relatively large number of superficially similar sentences, SBERT built on such a dataset is expected to be relatively robust to sentence pairs with high superficial similarity. To verify whether there is such a tendency, we partition STS Benchmark datasets in accordance with the superficial similarity of the sentences and investigate the per-

sentence 1	sentence 2	Human	Dice	w/oFT	SBERT	DefSent
A man is playing a guitar.	The man is playing the guitar.	4.909	0.800	0.906	0.985	0.978
A man is playing a guitar.	A guy is playing an instrument.	3.800	0.545	0.945	**0.646**	0.895
A man is playing a guitar.	A man is playing a guitar and singing.	3.200	0.833	0.979	0.874	**0.977**
A man is playing a guitar.	The girl is playing the guitar.	2.250	0.600	0.900	0.747	0.831
A man is playing a guitar.	A woman is cutting vegetable.	0.000	0.400	0.890	0.290	0.595

Table 2: Example sentence pairs in STS Benchmark datasets and their scores. "Human" denotes human-labeled similarity scores, "Dice" denotes Dice coefficients, and "w/oFT", "SBERT", and "DefSent" denote cosine similarities between each sentence embedding computed with BERT without fine-tuning, SBERT, and DefSent, respectively. The average cosine similarity for w/oFT is 0.816, for SBERT is 0.678, and for DefSent is 0.809.

formance of each embedding method on the partitioned datasets. Specifically, we use Dice coefficients between the sets of words in a sentence pair as the superficial similarity, which is defined as

$$\text{Dice}(S_1, S_2) = \frac{2|W_1 \cap W_2|}{|W_1| + |W_2|},$$

where S_1 and S_2 are the sentence pair, and W_1 and W_2 are the sets of words in $S1$ and $S2$, respectively. We sort the sentence pairs in all STS Benchmark datasets including training, development, and test sets in accordance with the Dice coefficient, and partition them into five subsets, that is, grouping 20% of the sentences from bottom to top.

Figure 3 shows the Spearman's ρ for each subsets. We can confirm that the subsets with larger Dice coefficients, that is, a higher superficial similarity, tend to be more difficult to rank the semantic similarities. However, as expected, SBERT is more robust to the subsets with higher superficial similarity, and consequently, SBERT achieves a higher score than DefSent for these subsets, whereas DefSent achieved a higher score than SBERT for the subsets with a lower superficial similarity.

For further investigation, we conduct a qualitative analysis of how superficial similarity affects the behavior of the methods. Table 2 shows example sentence pairs from STS Benchmark datasets with their human-labeled similarity scores, Dice coefficients, and cosine similarities between each sentence embedding with the respective methods. As shown in the second row from the top, we observe that each sentence of the pair represents almost the same thing except for minor details ("guitar" or "instrument"), but SBERT assigns relatively a much lower similarity than other examples. As shown in the third row from the top, the similarity score of DefSent is very high, even though the human-labeled score is not that high. In summary, we can say that SBERT is better at capturing the semantic similarity of superficially similar sentences,

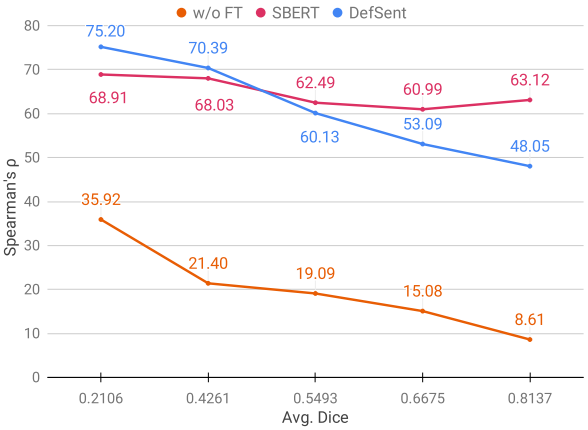

Figure 3: Spearman's $\rho \times 100$ for STS Benchmark partitioned in accordance with the ratio of shared words. Sentence pairs are more superficially similar to right.

while DefSent is better at capturing the similarity of sentences with low superficial similarity.

3.3 SentEval donwstream tasks

We then apply each type of embeddings to the downstream tasks of SentEval and analyze what type of information each type of embeddings captures that is useful for the downstream task. We train a logistic regression classifier with 10-fold cross-validation, a batch size of 64, an epoch size of 4, and Adam (Kingma and Ba, 2015) optimizer, the same as the default configurations of SentEval. Specifically, parameters of sentence embedding models are fixed during training of the classifier. We fine-tune and evaluate SBERT and DefSent three times with different seed values and report the average of accuracy for each downstream task. We also evaluate w/oFT for comparison.

Figure 4 shows the accuracy for downstream tasks. As a whole, SBERT and DefSent perform comparably. SBERT performs best for MR, CR, SST2, and MRPC. Since MR, CR, and SST2 are sentiment prediction tasks, it suggests that SBERT encodes the sentiment of sentences into the embedding. Also, MRPC is a paraphrase-prediction

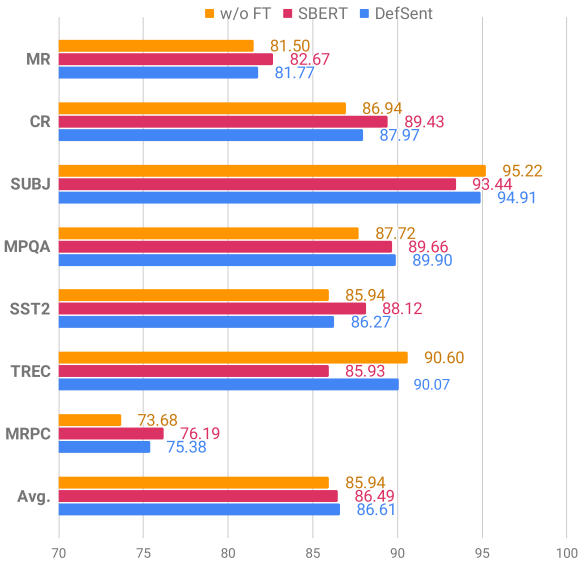

Figure 4: Experimental results on each SentEval downstream task with the accuracy (%).

task, which predicts whether two sentences have the same meaning on the basis of their embeddings. Therefore, MRPC is similar to the NLI task, and thus it is not surprising that SBERT performs better.

DefSent performs best for MPQA and is comparable to w/oFT for SUBJ and TREC. MPQA is a phrase-level opinion polarity classification task, and it is necessary to compose the meaning of phrases adequately. We conjecture that the performance of DefSent is high because DefSent successfully composes the meaning of the corresponding words from the definition sentences during fine-tuning. It is worth noting that w/oFT performs best for SUBJ and TREC, and SBERT performs much worse for them. SUBJ is a subjectivity classification task and TREC is a question-type classification task. Since information about words in sentences is particularly important for these tasks, SBERT is considered to have less information about which words are included in sentences than DefSent and w/oFT. Therefore, we can say that SBERT encodes mainly sentiment information into the sentence embedding, and the sentence embedding is suitable for determining whether the meaning is the same. Also, DefSent successfully composes the meaning of the sentence from its words and encodes information about words the sentence has.

3.4 SentEval probing tasks

Finally, we apply each type of embeddings to the probing tasks of SentEval and analyze what type of linguistic information each type of embeddings captures. We use the same setting as in Section 3.3.

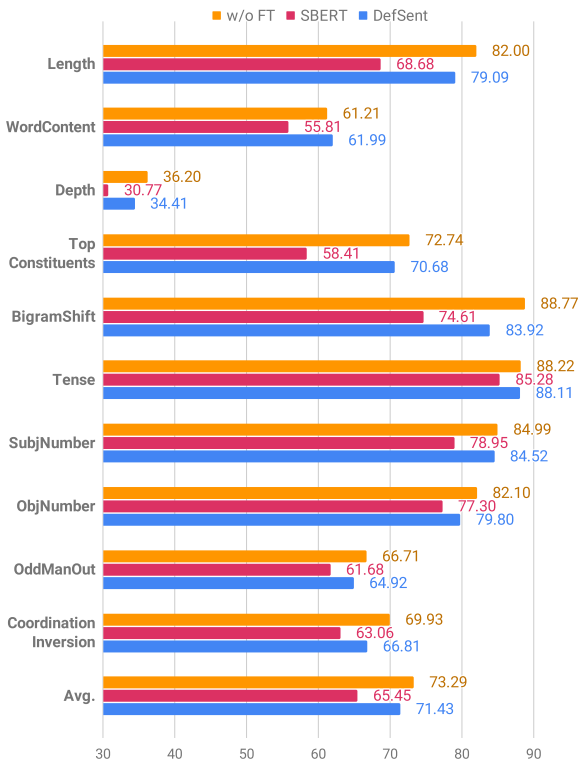

Figure 5: Experimental results on each SentEval probing task with the accuracy (%).

Figure 5 shows the accuracy for probing tasks. Overall, w/oFT performs best on average, followed by DefSent, and then SBERT. The overall performance of SBERT is relatively low. SBERT encodes the semantic information of sentences according to the results of SentEval downstream tasks. These results also indicate that SBERT encodes semantic information rather than linguistic information such as words in a sentence. DefSent is comparable to w/oFT in WordContent, Tense, and SubjNumber. This also indicates that the sentence embeddings from DefSent have information about words the sentence contains.

4 Combination of Sentence Embeddings

We have shown that SBERT and DefSent have different properties and that they may be complimentary. This suggests that combining the two methods may yield better sentence embeddings. Thus, we attempt to combine SBERT and DefSent and evaluate the resulting sentence embeddings on unsupervised STS tasks and SentEval downstream tasks. Specifically, we use the following five methods of combining SBERT and DefSent for BERT[2].

[2]The experimental results for RoBERT are given in Appendix C and D.

Model	Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT-base	w/oFT	30.88	59.90	47.74	60.29	63.73	47.29	58.22	52.58
BERT-base	SBERT	69.78	72.51	70.42	77.95	73.45	75.96	72.26	73.19
BERT-base	DefSent	67.31	81.76	71.83	78.18	76.91	76.98	73.47	75.20
BERT-base	S+D	70.71	**83.48**	**76.66**	**82.00**	**78.70**	**80.76**	**76.83**	**78.45**
BERT-base	D+S	68.68	73.65	70.60	76.96	72.54	75.30	72.46	72.89
BERT-base	MULTI	63.10	74.34	70.30	77.64	74.08	77.35	73.42	72.89
BERT-base	AVERAGE	**72.40**	81.36	75.80	81.90	77.64	79.74	75.87	77.81
BERT-base	CONCAT	71.13	78.54	74.03	79.95	76.01	78.37	74.17	76.03
BERT-large	w/oFT	27.69	55.78	44.48	51.67	61.85	47.00	53.85	48.90
BERT-large	SBERT	70.76	73.68	72.56	79.00	74.61	77.11	72.47	74.31
BERT-large	DefSent	63.30	82.16	72.67	79.06	77.52	77.40	74.02	75.16
BERT-large	S+D	69.48	**83.90**	76.83	**82.61**	**80.14**	**81.72**	**78.77**	**79.06**
BERT-large	D+S	71.25	75.71	73.39	79.68	75.20	77.67	73.78	75.24
BERT-large	MULTI	70.33	81.16	75.84	80.02	76.52	78.65	74.30	76.69
BERT-large	AVERAGE	**71.85**	82.60	**77.33**	82.52	79.12	80.71	76.30	78.63
BERT-large	CONCAT	71.37	80.28	76.08	81.10	77.63	79.57	74.71	77.25

Table 3: Experimental results on unsupervised STS tasks with Spearman's $\rho \times 100$.

S+D Fine-tuning the pre-trained model with SBERT then with DefSent sequentially.

D+S Fine-tuning the pre-trained model with DefSent then with SBERT sequentially.

MULTI Multi-task learning with SBERT and DefSent. The ratio of the size of the NLI dataset to the dictionary dataset is about 19:1, so we do 19 steps with SBERT and then 1 step with DefSent for the same model.

AVERAGE Averaging embeddings of separately fine-tuned models with SBERT and DefSent.

CONCAT Concatenate embeddings of separately fine-tuned models with SBERT and DefSent.

4.1 Evaluation on unsupervised STS tasks

We first estimate the resulting sentence embeddings on unsupervised STS tasks. We use the same settings described in Section 2.5. We use STS12–STS16, STS Benchmark test set (STS-B), and SICK-Relatedness (SICK-R) for the evaluation. We compute sentence similarities by using the cosine similarity of sentence embeddings derived from the respective combinations and calculate Spearman's ρ with gold labels. We conduct fine-tuning and evaluations 10 times with different seed values and report the average.

Table 3 shows the experimental results. The combinations S+D, AVERAGE, and CONCAT always outperform SBERT and DefSent. Among them, S+D achieves the best average score for base and large models. However we cannot confirm much performance improvement with D+S and MULTI. We leave an analysis of what affects this difference in performances as future work.

4.2 Evaluation on the SentEval tasks

We then estimate the resulting sentence embeddings on the SentEval tasks. We use the same settings described in Section 3.3. We conduct fine-tuning and evaluations three times with different seed values and report the average.

Table 4 shows the results. We can see that CONCAT achieves the highest average score but it should be noted that since SentEval performed supervised learning of a logistic regression classifier, the high dimensionality of the sentence embeddings of CONCAT is advantageous. Other than CONCAT, AVERAGE performs relatively well, which always outperforms S+D, D+S, and MULTI, unlike in the STS tasks. This suggests that fine-tuning the same model with different tasks might degrade the generalization ability.

5 Related work

Sentence embedding has been studied intensively. Kiros et al. (2015) proposed SkipThought, which trains a sentence embedding model by predicting the previous and next sentence from the embedding of a given sentence. Conneau et al. (2017) proposed InferSent, which trains a sentence embedding model built on BiLSTM in a Siamese network architecture on the NLI task. Cer et al. (2018) proposed Universal Sentence Encoder (USE), which is trained on an NLI dataset, and has also shown the effectiveness of NLI datasets in obtaining sophisticated sentence embeddings.

Recently, methods that leverage pre-trained language models to acquire sentence embeddings have attracted much attention. Pre-trained language models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), acquire linguistic

Model	Method	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
BERT-base	w/o FT	81.50	86.94	**95.22**	87.72	85.94	90.60	73.68	85.94
BERT-base	SBERT	82.67	89.43	93.44	89.66	88.12	85.93	76.19	86.49
BERT-base	DefSent	81.77	87.97	94.91	89.90	86.27	90.07	75.38	86.61
BERT-base	S+D	81.29	89.10	93.99	90.09	86.69	89.33	77.08	86.80
BERT-base	D+S	82.43	89.22	93.24	90.16	**88.98**	83.33	75.27	86.09
BERT-base	MULTI	81.73	88.80	93.17	89.27	87.28	87.87	75.54	86.23
BERT-base	AVERAGE	83.17	89.50	94.67	90.35	88.50	89.67	76.41	87.47
BERT-base	CONCAT	**83.24**	**89.64**	95.18	**90.51**	88.94	**90.60**	**77.37**	**87.93**
BERT-large	w/o FT	84.30	89.16	**95.60**	86.65	89.29	**91.40**	71.65	86.86
BERT-large	SBERT	84.76	90.61	94.08	90.04	90.77	85.47	75.90	87.38
BERT-large	DefSent	84.54	89.40	95.55	90.04	89.49	88.73	74.82	87.51
BERT-large	S+D	84.01	90.49	95.07	90.50	90.35	90.20	75.61	88.03
BERT-large	D+S	84.55	90.68	93.46	90.22	90.21	84.73	75.01	86.98
BERT-large	MULTI	84.63	90.56	94.10	89.85	90.23	88.70	76.56	87.80
BERT-large	AVERAGE	85.46	**90.92**	95.20	90.53	91.27	88.27	**77.00**	88.38
BERT-large	CONCAT	**85.53**	90.83	95.27	**90.66**	**91.95**	89.60	75.88	**88.53**

Table 4: Experimental results on each SentEval task with the accuracy (%).

knowledge by training on large texts and perform well on downstream tasks. Pre-trained models are also considered helpful for sentence embedding. There are two types of methods based on pre-trained models: unsupervised and supervised.

Unsupervised methods do not require labeled text but exploit the properties of pre-trained language models or create training data artificially. Li et al. (2020) showed that the sentence embedding space of BERT is anisotropic, and proposed BERT-flow, which learns a map to an isotropic Gaussian distribution to obtain sentence embedding. Several studies have also been based on contrastive learning, and are different in the way to make positive examples: DeCLUTR (Giorgi et al., 2021) takes into account different spans of the same document as positives; ConSERT (Yan et al., 2021) takes into account a pair of an original sentence and a collapsed sentence as positives; unsupervised SimCSE (Gao et al., 2021) takes into account the corresponding embeddings of the same sentence with different dropout masks applied as positives.

Supervised methods use labeled text to encode higher-level semantic information. Supervised methods generally produce more sophisticated sentence embeddings than unsupervised methods. In addition to SBERT and DefSent, supervised SimCSE (Gao et al., 2021) is one of the supervised sentence embedding methods. Supervised SimCSE fine-tunes BERT by contrastive learning using entailment pairs in the NLI datasets as positives.

6 Conclusion

In this paper, we empirically investigated the influence of supervision signals used for obtaining sentence embeddings. We focused on two methods: SBERT, which uses NLI datasets, and DefSent, which uses word dictionaries. We showed that there is a difference in the ability to order the similarity of sentences depending on their source or superficial similarity by comparing their performances on subsets of the STS datasets and tasks of SentEval. We found that SBERT is suitable for superficially similar sentence pairs because SBERT is based on the NLI datasets that contain a relatively large number of superficially similar sentences, whereas DefSent is suitable for sentence pairs that need to represent the compositional meaning because DefSent is based on definition sentences of a dictionary.

We also showed that SBERT performed better in tasks where sentiment information was important, while DefSent performed better in tasks where information about words and the compositionality of meaning were important by comparing their performances on downstream and probing tasks of SentEval. Finally, we demonstrated that combining the two methods yielded substantially better performance than the respective methods on unsupervised STS tasks and downstream tasks of SentEval.

For future work, we will expand the scope of our analysis to other pre-trained language models and sentence embedding methods to obtain insights for better sentence embeddings. In addition, We will investigate how those combination methods affect the properties of resulting sentence embeddings and explore how to effectively combine unsupervised sentence embedding methods, which have recently achieved good performance, such as DeCLUTR (Giorgi et al., 2021) and unsupervised SimCSE (Gao et al., 2021), with supervised sentenece embedding methods. Moreover, the combination of unsupervised methods, which

have recently achieved good performance, such as DeCLUTR (Giorgi et al., 2021) and unsupervised SimCSE (Gao et al., 2021), and supervised methods should also be promising.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 21H04901.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Semantic Evaluation (SemEval)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 32–43.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic Re-tuning with Contrastive Tension. In *International Conference on Learning Representations (ICLR)*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval)*, pages 1–14.

Daniel Matthew Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, C. Tar, Yun-Hsuan Sung, B. Strope, and R. Kurzweil. 2018. Universal Sentence Encoder. *arXiv:1803.11175*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, pages 1699–1704.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 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 (EMNLP)*, pages 670–680.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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 (NAACL)*, pages 4171–4186.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 879–895.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 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 (NAACL-HLT)*, pages 1367–1377.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)*.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3294–3302.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations (ICLR)*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 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)*, pages 216–223.

Nils Reimers and Iryna Gurevych. 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)*, pages 3982–3992.

Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. DefSent: Sentence Embeddings using Definition Sentences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 411–418.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008.

Bin Wang and C.-C. Jay Kuo. 2020. SBERT-WK: A Sentence Embedding Method by Dissecting BERT-Based Word Models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2146–2157.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning. *arXiv:2104.06979*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1112–1122.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 38–45.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 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 (ACL-IJCNLP)*, pages 5065–5075.

A Training Details

For fine-tuning of SBERT and DefSent, we use a batch size of 16, an epoch size of 1, Adam (Kingma and Ba, 2015) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a linear learning rate warm-up over 10% of training steps for each, as the same setting as Reimers and Gurevych (2019) and Tsukagoshi et al. (2021). We choose the learning rate that achieves the highest average score on the validation set for each respective model by fine-tuning three times with different seed values at each learning rate in a range of $x \times 10^{-6}, x \in \{1, 2, 5, 10, 20, 50\}$. We also use smart batching, and the max sequence length is 128 for training efficiency.

B Average Runtime and Computing Infrastructure

Fine-tuning of SBERT with BERT-base and RoBERTa-base took about 120 minutes on a single NVIDIA GeForce GTX 1080 Ti. Fine-tuning of DefSent with BERT-base and RoBERTa-base took about 10 minutes on a single NVIDIA GeForce GTX 1080 Ti. Fine-tuning of SBERT with BERT-large and RoBERTa-large took about 130 minutes on a single Quadro GV100. Fine-tuning of DefSent with BERT-large and RoBERTa-large took about 15 minutes on a single Quadro GV100.

C The details of evaluation on unsupervised STS tasks of RoBERTa

Table 5 shows the average of Spearman's rho for RoBERTa-base and RoBERTa-large on unsupervised STS tasks.

D The details of evaluation on SentEval of RoBERTa

Table 6 shows the average of accuracy for RoBERTa-base and RoBERTa-large on SentEval.

Model	Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
RoBERTa-base	w/o FT	30.61	55.55	46.78	58.43	61.21	54.36	62.17	52.73
RoBERTa-base	SBERT	70.20	74.44	71.86	78.70	74.47	76.92	72.11	74.10
RoBERTa-base	DefSent	60.05	76.16	69.06	74.07	77.86	76.58	74.05	72.55
RoBERTa-base	S+D	73.19	83.86	77.45	83.32	78.88	80.67	76.97	79.19
RoBERTa-base	D+S	70.97	75.07	72.50	79.04	74.56	77.13	72.81	74.58
RoBERTa-base	MULTI	69.27	77.34	73.10	80.68	76.08	77.97	73.61	75.44
RoBERTa-base	AVERAGE	71.61	78.65	74.65	80.30	76.71	78.56	74.04	76.36
RoBERTa-base	CONCAT	70.69	76.03	72.92	79.08	75.34	77.50	72.73	74.90
RoBERTa-large	w/o FT	26.00	54.35	44.10	56.35	60.37	47.01	58.11	49.47
RoBERTa-large	SBERT	74.04	79.47	75.47	82.77	79.50	80.49	74.19	77.99
RoBERTa-large	DefSent	57.79	74.67	69.01	72.98	75.48	77.39	72.55	71.41
RoBERTa-large	S+D	66.62	79.60	75.81	77.91	78.45	80.46	77.45	76.61
RoBERTa-large	D+S	74.18	79.81	76.38	82.85	78.78	80.38	74.86	78.18
RoBERTa-large	MULTI	61.34	57.43	60.17	75.56	73.78	74.92	70.10	67.62
RoBERTa-large	AVERAGE	73.43	82.97	77.85	83.82	80.65	82.09	75.91	79.53
RoBERTa-large	CONCAT	74.04	80.96	76.60	83.20	80.33	81.24	74.77	78.73

Table 5: Experimental results on unsupervised STS tasks with Spearman's $\rho \times 100$.

Model	Method	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
RoBERTa-base	w/o FT	84.35	88.19	95.28	86.49	89.46	93.20	74.20	87.31
RoBERTa-base	SBERT	85.35	91.50	93.15	90.95	92.06	87.07	76.62	88.10
RoBERTa-base	DefSent	84.70	91.15	94.55	90.56	89.88	92.40	76.43	88.52
RoBERTa-base	S+D	85.04	91.40	94.17	90.81	90.63	92.00	77.14	88.74
RoBERTa-base	D+S	85.20	91.34	93.45	90.84	92.20	88.20	76.29	88.22
RoBERTa-base	MULTI	85.15	91.00	93.25	90.69	91.47	89.67	77.08	88.33
RoBERTa-base	AVERAGE	85.57	91.66	94.01	91.14	92.55	89.67	78.12	88.96
RoBERTa-base	CONCAT	86.04	91.68	94.70	91.02	92.40	93.93	78.24	89.72
RoBERTa-large	w/o FT	85.46	88.72	96.04	88.34	91.27	93.80	73.80	88.20
RoBERTa-large	SBERT	87.35	92.56	94.13	90.99	92.77	92.20	76.00	89.43
RoBERTa-large	DefSent	86.28	91.14	95.12	90.97	90.74	92.33	73.74	88.62
RoBERTa-large	S+D	86.77	92.28	94.68	91.22	91.98	92.60	77.51	89.58
RoBERTa-large	D+S	87.02	92.40	93.62	90.80	92.59	90.93	77.35	89.25
RoBERTa-large	MULTI	87.52	92.56	94.39	91.09	93.15	91.60	76.69	89.57
RoBERTa-large	AVERAGE	87.82	92.81	94.69	91.36	93.24	93.93	77.49	90.19
RoBERTa-large	CONCAT	87.87	92.84	95.22	91.64	93.06	94.27	76.23	90.16

Table 6: Experimental results on each SentEval task with the accuracy (%).