Supervised Sentence Representation with Interactive Loss Decay

Anonymous EMNLP submission

Abstract

Contrastive learning has achieved remarkable results in sentence representation, but its semantic representation remains independent in the process of training and inference, and could not pay attention to the interactive information of sentence pairs. This paper proposes InterCSE, a interactive contrastive learning method for sentence embedding, which not only focuses on the semantic similarity sorting of sentence pairs, but also increases the interactive information as a supplement for sentence representation. Meanwhile, we propose a loss decaying strategy to balance embedding similarity and interactive information objectives. We evaluate the performance of InterCSE on standard semantic textual similarity (STS) tasks, and experiments show that our model using $BERT_{base}$ and $BERT_{large}$ achieve 82.11% and 82.88% spearman's correlation, 0.54% and 0.43% improvement compared to SimCSE respectively. We also conduct experiments that adding interactive network based on Sentence-Transformers, which get 85.18%(+0.88% $BERT_{base}$) and 85.69%(+1.07% $RoBERTa_{base}$) spearman's correlation on STS Benchmark. Hence, adding interactive features to the traditional siamese network performs very well, and achieves new state-of-the-art performance on sentence representation tasks.

1 Introduction

Sentence representation learning is a vital component of natural language processing tasks (Cer et al., 2017). The rapid development of sentence representation technology has made a wide range of downstream tasks more intelligent, especially information retrieval and text clustering.

Recently, the pre-trained language model has become the cornerstone of natural language processing technology, such as BERT(Devlin et al., 2019; Liu et al., 2019), GPT(Radford et al., 2018, 2019; Brown et al., 2020), ERNIE(Sun et al., 2019a,b, 2021), which greatly affects the development of various downstream tasks. Many sentence representation approaches are transformed into point-wise classification tasks(Reimers and Gurevych, 2019), and there is a gap between the training optimization objectives and good sentence representation. Sentence representation could not get high performance directly with pre-trained language model because of anisotropic phenomenon (Gao et al., 2019), but contrastive learning play the role of a bridge. Comparative learning applys the sorting method with InfoNCE(van den Oord et al., 2019) contrastive loss function, which is more suitable for the optimization goal of sentence representation.

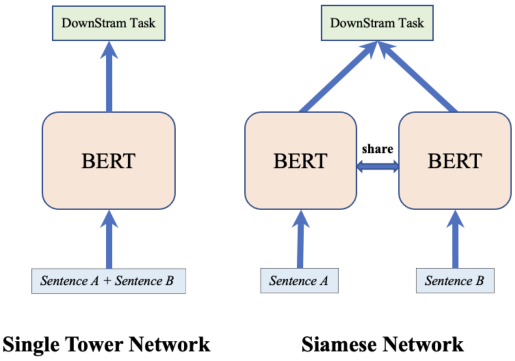

Figure 1: The architecture of single tower network(left) and siamese network(right).

The sentence representation model(Gao et al., 2022; Yan et al., 2021) of contrastive learning combined with pre-trained model is a siamese network, which encodes all sentences independently. The structure of the siamese network makes it possible to quickly produce the vector representation of the sentences and calculate the similarity during training and prediction, so as to complete the retrieval or sentence clustering task in massive data. However, the structure of independent encoding makes the sentences pair lose the interactive information, which reduces the accuracy rate.

Network Type	Model	Spearman
Siamese Network	SBERT-base ♡	84.7
	SBERT-large ♡	84.5
	SimCSE-BERT-base ♠	84.3
	SimCSE-BERT-large(reproduce)	85.4
	SRoBERTa-base ♡	84.9
	SRoBERTa-large ♡	85.0
	SimCSE-RoBERTa-base ♠	85.8
	SimCSE-RoBERTa-large ♠	86.7
Single Tower Network	BERT-base ♦	85.8
	BERT-large ♦	86.5
	RoBERTa-base ♢	87.2
	RoBERTa-large ♢	88.1

Table 1: Comparison of singel tower network and siamese network on STS-Benchmark(Cer et al., 2017) test set with spearman's correlation. ♡: results from Sentence-Transformers(Reimers and Gurevych, 2019), ♠: results from SimCSE(Gao et al., 2022), ♦ : results from BERT (Devlin et al., 2019), ♢: the performance of RoBERTa-base and RoBERTa-large in single tower network are reproduce through fairseq(Ott et al., 2019). SimCSE models are trained on NLI datasets(Bowman et al., 2015), and the other models are trained on STS-Benchmark train set. All results are rounded to one decimal place for comparison.

There are obvious differences between single tower network and siamese network. The network structure is depicted in Figure 1. The single tower network is like the original BERT(Devlin et al., 2019) model structure. After concating the sentence pairs, the semantic features of the sentence pairs are extracted and input to the downstream classification or regression network. The siamese network(Koch et al., 2015) has two encoders that encode each sentence individually where the two encoders share model parameters. After encoding, it can perform similarity calculation or put into the downstream network using sentence embedding, such as Sentence-Transformers(Reimers and Gurevych, 2019).

The performance of the single-tower network and the siamese network on the STS Benchmark test dataset is shown in Table 1. On the whole, the single tower network has a good improvement in spearman's correlation index compared with the siamese network. We consider that when the single tower network encodes a sentence pair, the sentence is not an independent individual. It will refer to its counterparts for encoding and use the attention mechanism to extract features, which makes the single tower network in the sentence pair similarity task has a higher correlation coefficient. Recently, most sentence representation tasks use siamese networks. Although it could bring computational advantages on massive data, it inevitably reduces the accuracy of correlation.

In order to take full advantage of the accuracy advantages of single tower networks and the inference speed advantages of siamese networks, this paper proposes a multi-task contrastive learning method. On the basis of SimCSE, we added a single tower network as a supplement to form a framework for multi-task learning. This method could fully obtain the interaction information between sentence pairs during the process of training sentence representation.

Our contributions can be summarized as follows:

1. We propose an effective framework of multi-task contrastive learning for sentence representation tasks which could introduce sentence interactive semantic information.

2. We design a loss function for the proposed framework. When the interactive network introduce information increment, try to minimize the hurt to the original sentence representation.

3. Experiments show that our approach achieves new state-of-the-art performance on STS tasks.

4. We also introduce sentences interactive network based on Sentence-Transformers, and get a quite outstanding performance on STS benchmark task.

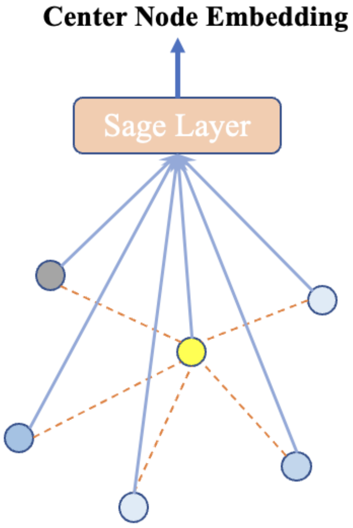

Figure 2: The architecture of GraphSage. Sage layer aggregates all neighbor nodes' embedding for center node.

2 Related Work

2.1 Language Model

Recently, transformer(Vaswani et al., 2017) structure shines in the field of deep learning and lays the foundation for large models. The attention mechanism is good at capturing the semantic relationship between sequences. Using transformer's encoder, many language models have been born, such as GPT(Radford et al., 2018), BERT(Devlin et al., 2019), RoBERTa(Liu et al., 2019), XLNet(Yang et al., 2019), Ernie(Sun et al., 2019a), etc. These models mainly have some differences in masking mechanism, pre-training data and methods. Among them, the BERT model mainly masks 15% words in the sentence and predicts the origin words as a unsupervised pre-training task. After obtaining the pre-trained model, NLP downstream tasks only need to complete fine-tuning on the model to achieve high performances, including sentence classification, sequence tagging, question answering, machine reading and comprehension and sentence-pair classification or regression.

2.2 Contrastive Learning and Sentence Representation

Contrastive learning was first proposed in the field of computer vision. It mainly wants to optimize the similarity of sample pairs in a representation space, and make similar sample pairs gather and dissimilar sample pairs stay away. After the sample representation is obtained using siamese network embedding, the sample similarity is calculated to maximize the similarity of the positive samples. Contrastive learning could take a cross-entropy objective with in-batch-negatives. This optimization scenario is very suitable for unsupervised scenarios. Usually, positive samples can be obtained through simple data enhancement, and a large number of negative samples can be obtained through negative sampling.

Type of NLI datasets	Numbers
Entailment	314k
Neutral	314k
Contradiction	314k
Entailment+Contradiction	270k

Table 2: Statistics of different type of SNLI+MNLI datasets.

In terms of sentence representation, the idea of contrastive learning continues to be used, leading to many research results, such as ConSERT(Yan et al., 2021), SimCSE(Gao et al., 2022) and ESimCSE(Wu et al., 2022). ConSERT proposes four different data augmentation strategies to generate views for contrastive learning, including adversarial attack, token shuffling, cutoff and dropout. SimCSE first describes an unsupervised approach, which takes an input sentence and predicts itself in a contrastive objective, with only standard dropout used as noise. Only using dropout as data augmentation becomes a popular method for contrastive learning. ESimCSE introduces two modifications based on SimCSE. It applies a simple repetition operation to modify the input sentence, and then passes the input sentence and its modified counterpart to the pre-trained Transformer encoder, respectively, to get the positive pair. And it introduces a momentum contrast, enlarging the number of negative pairs.

2.3 Multi-Task Learning

Multi-Task Learning (MTL) is an important research topic in machine learning, which aims to learn multiple tasks simultaneously. In MTL, multiple tasks are divided into multiple learning units, and a learner can learn multiple tasks by making multiple small models.

One of the important research directions in MTL is to develop efficient algorithms and theoretical models. In recent years, many works have been

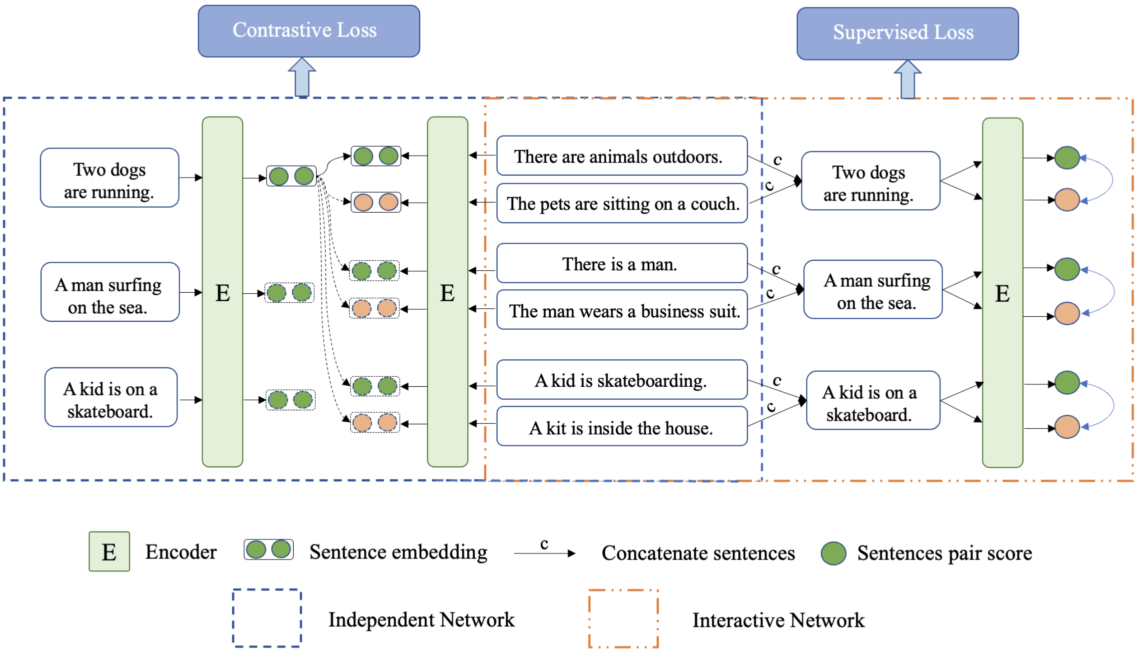

Figure 3: The architecture of InterCSE. Independent network focuses on the representation of sentence, and interactive network perceives sentences interactive semantic. Independent network encodes sentences individually, while interactive network concatenates two sentences and predicts the degree of semantic relevance. Three encoders share same parameters when training.

carried out in this direction, including deep neural networks, attention mechanism, joint attention, graph attention, and so on. MTL has been successfully used in natural language processing applications, including text classification(Liu et al., 2017), machine translation(Luong et al., 2016), sequence labeling(Rei, 2017) and sentence representations(Ahmad et al., 2018). These algorithms and theoretical models can help MTL learn multiple tasks with better performance.

3 InterCSE

In this section, we present InterCSE (introducing **Inter**active semantic information in **C**ontrastive **S**entences **E**mbedding) for sentence representation task. Firstly, we present the overall framework of our approach. Then, we introduce the training dataset for our model. Finally, we talk about the combination strategies of loss function for multiple objectives.

3.1 Framework

Our approach is mainly inspired by Graph-Sage(Hamilton et al., 2018). Transformer-Sage layer aggregates the embedding of neighbor nodes, and realizes an information interaction between neighbors, which enhances the representation ability of center nodes, as shown in Figure 2.

The main idea of our approach is to introduce interactive information while encoding a sentence. Hence, a single tower network which we call interactive network in InterCSE is added on the basis of SimCSE(Gao et al., 2022) as shown in Figure 3. There are two major components in our framework: independent network and interactive network. During the training process, the interactive network completes the interactive feature extraction of sentence pairs. While inference, only independent network works and generates sentence embedding. All the transformer encoders share same parameters.

Independent Network The goal of the independent network is to quickly produce semantic representation of single sentence without relying on other sentences. The independent network is composed of a transformer encoder and cosine network. Transformer encoder is a siamese network based on BERT-like model, and cosine network is generally cosine similarity calculation. The input is exactly two sentence, such as sentence A and sentence A^+ (or sentence A^-)[1], and independent

[1] For convenience of description, sentence A refers *Two dogs are running.*, sentence A^+ refers *There are animals outdoors.*, and sentence A^- refers *The pets sitting on a couch.* in Figure 3.

network encodes the sentence pairs independently by *[CLS]* representation, then calculates the cosine similarity. There are a high semantic score between sentence A and sentence A^+, and a low semantic score between sentence A and sentence A^-.

Interactive Network The goal of the interactive network is to obtain non-independent sentence semantic information through the attention interaction of words between sentence pairs. Non-independent sentences semantic information is an important correlation signal, which can keenly perceive the semantics of sentence to details. The interactive network consists of an transformer encoder and a feed forward network. Transformer encoder is a BERT-like model, and the input is the concatenation of two sentence and *[SEP]* token. The *[CLS]* embedding of transformer encoder is input into the feed forward network to evaluate the similarity of sentence pairs or classification. For example, concatenation of sentence A and sentence A^+ is positive, concatenation of sentence A and sentence A^- is negative.

3.2 Training Dataset

The model we proposed is suitable for supervised tasks, and the in-batch-negative sampling method of the independent network requires discrete label data, so we introduce natural language inference (NLI) datasets to train our model, including the SNLI(Bowman et al., 2015) and MNLI(Williams et al., 2018) datasets. NLI datasets consist of high-quality pairs, and given a premise, human annotators generate three types of sentences: entailment(is absolutely true), neutral(might be true), and contradiction(is definitely false).

Following the work of SimCSE(Gao et al., 2022), we adopt the hard negative strategy, using entailment and contradiction to represent positive and negative samples, respectively. Therefore, a triplet is generated, (x_i, x_i^+, x_i^-), where x_i is the premise, x_i^+ and x_i^- are entailment and contradiction hypotheses. Statistics of training datasets are shown in Table 2.

3.3 Loss Expression

The model we proposed is a multi-task model, including two modules of interactive network and independent network. The modeling objectives of the two modules are different, and the corresponding loss functions are also different. The following describes the loss functions corresponding to the two modules in detail.

For the independent encoding module, the input is a mini-batch of data, we take a cross-entropy objective with in-batch negative sampling. We also use hard negative in our model, and the contrastive loss $loss_{cl}$ can be expressed as:

$$-log\frac{e^{sim(h_i,h_i^+)/\tau}}{\sum_{j=1}^{N}(e^{sim(h_i,h_i^+)/\tau}+e^{sim(h_i,h_i^-)/\tau})} \quad (1)$$

where sim$(\cdot)$ indicates cosine similarity function, τ controls the temperature, h_i, h_i^+ and h_i^- is the embedding of x_i, x_i^+ and x_i^- respectively.

For the interaction module, the input is a mini-batch data where there are N pairs (x, x^+) and (x, x^-), and the label of (x, x^+) and (x, x^-) are 1 and 0 respectively. We consider two kind of loss functions to perceive the interactive semantic information.

The first is classification loss for all sentences in a mini-batch, and we use cross entropy loss $loss_{ce}$ to express as:

$$-\sum_{i=1}^{2N}(y_i log(p_i) + (1-y_i)log(1-p_i)) \quad (2)$$

where y_i is the label, p_i is the predicted score of feed forward network.

The second is margin ranking loss for the ranking relation between positive and negative samples in a mini-batch. Margin ranking loss $loss_{mr}$ could enhance the pairwise distinction in semantic space, and it can be express as:

$$\sum_{i=1}^{N} max(0, margin - (p_i^+ - p_i^-)) \quad (3)$$

where p_i^+ and p_i^- is the predicted score of feed forward network of (x_i, x_i^+) and (x_i, x_i^-), $margin$ controls the boundary.

With classification loss and margin ranking loss, we get supervised loss for interactive network as:

$$loss_{sl} = loss_{ce} + loss_{mr} \quad (4)$$

Multi-task learning needs to integrate different loss functions. We propose the weighted decaying methods to regularize three different loss functions:

While capturing the interactive semantic information, we need to minimize the hurt to siamese encoder. We propose the loss decaying methods to regularize two different loss functions as:

$$loss = loss\_cl + \alpha * loss\_sl \quad (5)$$

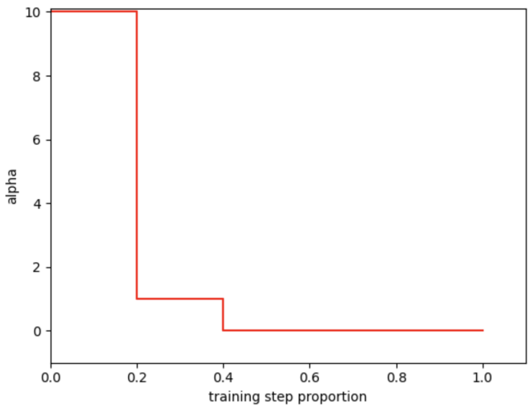

Figure 4: α piecewise constant decaying when training.

where α is piecewise constant decaying when training. For example, we evenly divide the training process into 5 stages, each stage corresponds to a different α. When the value list of α is $[10, 1, 0.1, 0.01, 0.001]$, the decaying is shown on Figure 4.

4 Experiments

Our approach is mainly proposed for supervised tasks, and we conducted multiple experiments on Semantic Textual Similarity (STS) task to verify the effectiveness of this approach.

4.1 Setups

Datasets Following previous works(Reimers and Gurevych, 2019; Gao et al., 2022; Yan et al., 2021), we evaluate our approach on 7 STS tasks: STS 2012-2016(Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark(Cer et al., 2017) and SICK-Relatedness(Marelli et al., 2014). A pair of sentences in those datasets has a gold score between 0 and 5 to indicate their semantic similarity. The higher the score, the higher the similarity of sentences pair.

Since the gold score label of the STS datasets is not suitable for the training process of the independent network, we introduce the SNLI(Bowman et al., 2015) and MNLI(Williams et al., 2018) (NLI) datasets as supervised data to train our model. The details of NLI datasets have described in Section 3.2.

Evaluation When evaluating the trained model, we first obtain the representation of sentences by *[CLS]* token embeddings, then we report the spearman correlation between the cosine similarity scores of sentence representations and the human-annotated gold scores. When calculating spearman correlation, we merge all sentences together in a STS task, and calculate the mean of spearman correlation.

4.2 Training Details

Our implementation is based on the SimCSE. We start from pre-trained checkpoints of BERT-base(uncased) and BERT-large (uncased), take the *[CLS]* representation as the sentence embedding, and train model on the combination of MNLI and SNLI datasets on 4 Telsa V100 gpus for 4 epochs. Hyper-parameters τ and $margin$ are set to 0.04 and 0.5. Since α is used to learn interactive information and could not hurt performance of sentence embedding, the value is chose from $[10, 1, 0.1, 0.01, 0.001]$ following with the training process.

4.3 Main Results

We compare our approach to previous state-of-the-art sentence embedding methods on STS tasks, including InferSent(Conneau et al., 2017), Universal Sentence Encoder(Cer et al., 2018), Sentence-BERT(Reimers and Gurevych, 2019), ConSERT(Yan et al., 2021), SimCSE(Gao et al., 2022) and PromCSE(Jiang et al., 2022a).

Table 5 shows the evaluation results on 7 STS tasks. InterCSE can substantially improve results on all the datasets, greatly outperforming the previous state-of-the-art models. Specifically, InterCSE-BERT(base) improves the averaged spearman's correlation from 81.57% to 82.11% compared to SimCSE-BERT(base). InterCSE-BERT(large) achieve 82.88% spearman's correlation, 0.43% improvement compared to SimCSE-BERT(large). Our models achieve new state-of-the-art performance on STS tasks.

5 Inter-Sentence-Transformers

In this section, for demonstrating the generality of interactive information on sentence representation tasks, we propose to add interactive network to the Sentence-Transformers(Reimers and Gurevych, 2019), and conduct some experiments to verify the performance on the STS-Benchmark dataset.

5.1 Model and Loss Function

Our proposed model is called Inter-Sentence-Transformers[2]. It also consists of independent network and interactive network.

[2] https://github.com/2hip3ng/InterCSE/tree/main/Inter-Sentence-Transformers

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
InferSent-Glove ♣	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder ♣	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
SBERT(base) ♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT-flow(base) ♠	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
SBERT-whitening(base) ♠	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
ConSERT-joint(base) ◇	70.53	79.96	74.85	81.45	76.72	78.82	77.53	77.12
SimCSE-BERT(base) ♠	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
PromCSE-BERT(base)△	75.58	84.33	79.67	85.79	**81.24**	84.25	80.79	81.81
InterCSE-BERT(base) ♡	**75.83**	**85.05**	**80.82**	**86.00**	81.07	**84.86**	**81.16**	**82.11**
SBERT(large) ♣	72.27	78.46	74.90	80.99	76.25	79.23	73.75	76.55
SimCSE-BERT(large) ♦	76.15	86.29	80.80	86.27	81.29	85.39	80.98	82.45
InterCSE-BERT(large) ♡	**76.59**	**86.69**	**81.83**	**86.32**	**81.85**	**85.78**	**81.12**	**82.88**

Table 3: Sentence embedding performance on STS tasks (Spearman's correlation, "all" setting). ♡: results of our approach, ♣: results from sentence-transformer(Reimers and Gurevych, 2019), ◇: results from ConSERT(Yan et al., 2021), ♠: results from SimCSE(Gao et al., 2022), △: results from PromCSE(Jiang et al., 2022b), ♦ : results from our reproduced.

The independent network follows Sentence-Transformers with siamese transformers, and uses the mean of all tokens embedding as sentences representation. The predicted score produced by cosine-similarity on sentences representation is called $indep\_score$.

The interactive network extracts the *[CLS]* token representation as input of a layer of regression network which is made up of fully-connected layer and sigmoid function. The predicted score produced by sigmoid function is called $inter\_score$.

Interactive and independent network both use mean-square error as loss function, and we combine these to generate a final loss with piecewise-constant decay as shown in equation (6) to (8).

$$L_{indep} = \frac{1}{N}\sum_{i=1}^{N}(y_i - indep\_score_i)^2 \quad (6)$$

$$L_{inter} = \frac{1}{N}\sum_{i=1}^{N}(y_i - inter\_score_i)^2 \quad (7)$$

$$Loss = L_{indep} + \alpha * L_{inter} \quad (8)$$

where y_i is the true score of sentences pair, α is decaying as training.

5.2 Experiment Results

We conduct experiments on the STS-Benchmark dataset that use STS-Benchmark training dataset to train our model and evaluate on test dataset with spearman's correlation. The label of dataset is a number between 0 and 5, so we divide it by 5.0 as the two sentences semantic similarity.

Model	Spearman
SBERT-STSb-base ♣	84.30
SBERT-STSb-large ♣	84.28
SRoBERTa-STSb-base ♣	84.62
SRoBERTa-STSb-large ♣	84.41
Inter-SBERT-STSb-base ◇	**85.18**
Inter-SBERT-STSb-large ◇	**84.91**
Inter-SRoBERTa-STSb-base ◇	**85.69**
Inter-SRoBERTa-STSb-base ◇	**84.82**

Table 4: Evaluation on the STS Benchmark test set with spearman's correlation. ♣: reproduced on Sentence-Transformers(Reimers and Gurevych, 2019); ◇: adding interactive network based on Sentence-Transformers. All results are trained on STS Benchmark train set.

The piecewise constant value α is chose from [10, 1, 0.1, 0.01, 0.001] in BERT-like base models and [1, 0.1, 0.01, 0.001, 0.0001] in BERT-like large models. Training epochs is 32, and the other parameters are the same as Sentence-Transformer.

The final performance is shown in the table 4. Inter-Sentence-Transformers get a state-of-the-art performances. It improves 0.88% and 1.07% based on BERT-base and RoBERTa-base respectively compared with Sentence-Transformers. And it slight increases based on BERT-large and RoBERTa-large. Inter-Sentence-Transformers treats sentence representation as a regression task. Experiment results show that our interactive network plays an effective role for sentence representation.

Training Strategy	STS-B Spearman
Single-Siamese	37.74
Siamese-Single	84.76
InterCSE	**86.25**

Table 5: Evaluation on the STS Benchmark dev set with spearman's correlation of training strategies, and the initial checkpoint is BERT-base.

Model	STS-B	STS Avg.
Loss		
w/o $loss_{ce}$	86.19	82.02
w/o $loss_{mr}$	86.16	81.97
constant α	86.08	81.75
square penalty	86.09	81.76
decay α	**86.25**	**82.11**

Table 6: Evaluation on the STS Benchmark dev set with spearman's correlation of loss function.

α **decay list**	STS-B Spearman
$[100, 10, 1, 0.1, 0.01]$	86.19
$[10, 1, 0.1, 0.01, 0.001]$	**86.25**
$[1, 0.1, 0.01, 0.001, 0.0001]$	86.09
$[0.1, 0.01, 0.001, 0.0001, 0.00001]$	86.02

Table 7: Evaluation on the STS Benchmark dev set with spearman's correlation of α decay list.

6 Ablation Experiment

In this section, we conduct further analyses to understand the role of interactive network in InterCSE, including training strategy, loss combination and loss decay.

Training Strategy We think that there are three training strategies that can help the model to consider the interaction information of sentence pairs while capturing sentence independent semantics:

(a) *Single-Siamese*, we firstly train single tower network, and then train siamese network;

(b) *Siamese-Single*, we firstly train siamese network, and then train single tower network;

(c) *InterCSE*, we use joint training strategy with loss decay.

The performance of three training strategy are shown as Table 5, and the joint training strategy has a obvious advantage.

Loss Combination From the experimental results of InterCSE and Inter-Sentence-Transformers, we can see the effectiveness of the interaction network for improving the overall performance. At the same time, in the section *Training Stragegy*, the benefits of multi-task methods for training two subnetworks at the same time are also confirmed. And there are still many methods how to combine the loss of multi-tasking. Therefore, we have carried out a variety of combination strategies:

(a) remove $loss_{ce}$ from equation (4),

(b) remove $loss_{mr}$ from equation (4),

(c) set α as a constant value, and we use 1.0,

(d) add square penalty as below,

$$loss = loss\_cl^2 + loss\_sl^2 \qquad (9)$$

(e) use loss decay method.

The performance of different loss is shown in Table 6. Removing $loss_{ce}$ or $loss_{mr}$ lose some semantic supervision signal which lower overall performance, and loss decay get an awesome result than constant α and square penalty.

Loss Decay In our approach, we use a dynamically changing loss to express the multi-task learning goal, which does not affect the sentence semantic representation while taking more semantic information. Hence, we propose an attenuation strategy for the loss of the interactive network, as shown in the equation (5), and α gradually decreases with the training process going on. The value of α represents the degree of interactive information introduced. The larger the α, the more sufficient the interactive information introduced, but it will also affect the sentence representation of the original siamese network. The results of different descent methods we compared are shown in Table 7. When α decay list is $[10, 1, 0.1, 0.01, 0.001]$, our model get best performance. In the early stage of training, the interactive network has a large loss, which makes the model perceive the interactive information and update the model parameters by back-propagation. In the later stage of training, the interaction network is weakened, and the model fully learns the semantic embedding of the sentence.

7 Conclusion

In this paper, we propose multiple task method joint contrastive learning for sentence representation which is termed InterCSE. Experiments shows that InterCSE achieves considerable performance on 7 semantic text similarity tasks. Through InterCSE uses a simple framework, it makes a prefect combination of the advantages of single tower network and siamese network by decaying loss function. When perceiving fine-grained word information, our approach try to minimize damage to sentence semantics. Meanwhile, the performance of Inter-Sentence-Transformers is improved a lot than Sentence-Transformers, it proves that the interactive network can bring a good positive gain again. In the future, we could focus on designing reinforcement learning method with human feedback, which help to enhance the sentence embedding.

Limitations

One limitation of our work is that we can not experiment on unsupervised setting, though the interactive network of InterCSE and Inter-Sentence-Transformers needs labeled sentence pairs. In the unsupervised task of sentence representation, how to extract interaction information is still a tough challenge.

Acknowledgements

TODO

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 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

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 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

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-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

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 – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

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), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Wasi Uddin Ahmad, Xueying Bai, Zhechao Huang, Chao Jiang, Nanyun Peng, and Kai-Wei Chang. 2018. Multi-task learning for universal sentence embeddings: A thorough evaluation using transfer and auxiliary tasks.

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*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

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-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

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

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2018. Inductive representation learning on large graphs.

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022a. Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning.

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022b. Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3021–3035, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. 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.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning.

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'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019a. Ernie: Enhanced representation through knowledge integration.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019b. Ernie 2.0: A continual pre-training framework for language understanding.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Cite arxiv:1906.08237Comment: Pretrained models and code are available at https://github.com/zihangdai/xlnet.