CoSENT: Consistent Sentence Embedding via Similarity Ranking

Xiang Huang, Hao Peng, Dongcheng Zou, Zhiwei Liu, Jianxin Li, Kay Liu, Jia Wu,
Jianlin Su, Philip S. Yu *Fellow, IEEE*

***Abstract*—Learning the representation of sentences is fundamental work in the field of Natural Language Processing. Although BERT-like transformers have achieved new SOTAs for sentence embedding in many tasks, they have been proven difficult to capture semantic similarity without proper fine-tuning. A common idea to measure Semantic Textual Similarity (STS) is considering the distance between two text embeddings defined by the dot product or cosine function. However, the semantic embedding spaces induced by pretrained transformers are generally non-smooth and tend to deviate from a normal distribution, which makes traditional distance metrics imprecise. In this paper, we first empirically explain the failure of cosine similarity in semantic textual similarity measuring, and present CoSENT, a novel Consistent SENTence embedding framework. Concretely, a supervised objective function is designed to optimize the Siamese BERT network by exploiting ranked similarity labels of sample pairs. The loss function utilizes uniform cosine similarity-based optimization for both the training and prediction phases, improving the consistency of the learned semantic space. Additionally, the unified objective function can be adaptively applied to different datasets with various types of annotations and different comparison schemes of the STS tasks only by using sortable labels. Empirical evaluations on 14 common textual similarity benchmarks demonstrate that the proposed CoSENT excels in performance and reduces training time cost.**

***Index Terms*—sentence embedding, semantic textual similarity, similarity ranking, siamese network**

I. Introduction

LEARNING to represent text effectively is a fundamental task in the field of Natural Language Processing (NLP). In particular, representative and versatile sentence representation is crucial for numerous NLP tasks, including language translation [1], sentiment classification [2], [3], information retrieval [4], question-answering [5], etc. In recent years, a series of transformer-based [6] pre-trained models, represented by BERT [7], have achieved state-of-the-art performance on a variety of sentence representation tasks.

TABLE I: Directly optimizing the cosine similarity may lead to model training failure. ϵ is the threshold of contrastive loss. We report Spearman's rank correlation as $\rho \times 100$ between the cosine similarity of sentence representations and the gold labels on multiple datasets.

ϵ	PAWS [13]	STSb [14]	SICK-R [15]	ATEC [16]	BQ [17]
-1	6.52	65.88	59.10	6.83	63.48
-0.5	48.43	67.79	60.00	10.28	67.91
0	72.72	71.68	59.42	44.60	71.20
0.5	74.62	72.93	58.50	50.22	72.12
0.9	75.22	71.03	57.70	50.69	71.79

However, recent studies [8], [9], [10] observed that the representations from BERT are not uniformly distributed with respect to direction but are anisotropic, occupying a narrow cone in the vector space. This leads to the model failing to effectively measure the semantic similarity between texts without proper fine-tuning and, in some cases, even being inferior to simple embedding models like GloVe. To better exploit downstream supervision of text matching, InferSent [11] and Sentence-BERT (SBERT) [12] propose a fine-tuning paradigm based on the Siamese network architecture. Specifically, the Siamese network has two branches of parameter-shared encoders and ingests a pair of sentences. Then, the encoder is optimized via the similarity labels (e.g., -1 denoting dissimilar) of this pair of sentences. Despite the effectiveness of the Siamese network, the inconsistency scheme between training and prediction of SBERT may lead to potential issues. Firstly, since the training loss function is irrelevant to the cosine-based evaluation metric (e.g., Spearman rank correlation), the training process could potentially collapse, i.e., the performance score on the validation set significantly decreases rather than increases as the loss decreases on the training set. Secondly, though SBERT employs a softmax classifier on concatenated vectors during the training process, there is no detailed theoretical justification regarding the effectiveness. In addition, SBERT designs three loss functions for different types of mainstream text semantic matching datasets. However, there is no generic rule for selecting the best one, and those loss functions are inconsistent, which thus induces extra difficulties in practice.

To address the aforementioned issues, we focus on providing an explanation for the failure of cosine similarity measurement in most BERT-based textual semantics. In fact, a sentence pair labeled as negative may share partial similarity in the semantic space, e.g., they are different sentences but discuss the same topic. These sentence pairs are defined as *hard negative samples*, as they are more similar than

Xiang Huang, Hao Peng, and Dongcheng Zou are with the School of Cyber Science and Technology, Beihang University, Beijing 1000191, China. E-mail: {huang.xiang, penghao, zoudongcheng}@buaa.edu.cn;

Jianxin Li is with the School of Computer Science and Engineering, Beihang University, Beijing 1000191, China. E-mail:lijx@buaa.edu.cn;

Zhiwei Liu, Kay Liu, and Philip S. Yu are with the Department of Computer Science, University of Illinois Chicago, Chicago, IL 60607, USA. E-mail:{zliu213, zliu234, psyu}@uic.edu;

Jia Wu is with the School of Computing, Macquarie University, Sydney, Australia. E-mail: jia.wu@mq.edu.au;

Jianlin Su is with Zhuiyi Technology, Shenzhen, China. E-mail: bojonesu@wezhuiyi.com.

Manuscript received February 2023, minor revised April 2024, accepted May 2024. (Corresponding author: Hao Peng.)

the other negative samples. Hence, we claim that labels are insufficiently precise to reflect the semantic similarity between sentences, and directly optimizing cosine similarity with these inaccurate supervisions would lead to the low generalization ability of the model. To provide a more detailed explanation, we illustrate this concept by analyzing the contrastive loss. This loss function is designed to increase the proximity of sentences to positive sentence samples and to distance them from negative sentence samples if their cosine similarity exceeds the threshold ϵ (see Eq. 2). As shown in Table I, we can improve sentence matching performance simply by increasing ϵ to adapt the model to a dataset with more hard negative samples. For datasets with a high proportion of hard samples like PAWS [13] and ATEC [16], the model training performance can be significantly improved by increasing the threshold ϵ appropriately, as it maintains enough similarity space for the hard samples. Details of the experiment are presented in Section III-A. These results indicate that those negative samples in datasets retain semantic similarity.

In this sense, we propose a novel training paradigm, the CoSENT, which is a consistent ranking-based semantic matching framework for pre-trained language models. In light of the shortcomings of SBERT and cosine similarity, it is necessary to design a consistent and unified loss for semantic matching, which can optimize a uniform cosine similarity-based objective for both the training and prediction phases and utilize different types of supervision, labeling, or scoring for more flexible optimization. Through the analysis of cosine loss, we find that the distances manifested by labels (e.g., scores and positive-negative labels) cannot precisely reflect the actual similarity of sentence pairs in the semantic space, due to the absence of an objective reference in the manual labeling. Nevertheless, since the manual label is built upon the subconscious semantic comparison, the similarity ranking in annotations of sentence pairs still contains rich information (e.g., even for hard samples, the overall similarity of positive pairs is greater than that of negative pairs, albeit with small differences). Therefore, a novel loss function is presented that aims at maintaining the similarity ranking of the sentence pairs, which complements the cosine similarity without introducing extra parameters or network complexity. Concretely, for a sample batch sorted by annotated similarity, the optimization goal is to align the ranking of the cosine similarities between the sentence vectors of the sample pairs with the ranking of the annotated similarities, which flexibly improves the distinction of representation in the BERT semantic space.

To verify the effectiveness and adaptability of CoSENT, extensive and diverse experiments are conducted, including unsupervised/supervised Semantic Textual Similarity (STS) evaluation, embedding evaluation on transfer tasks, interaction-based STS evaluation, convergence speed evaluation, and the detailed study (e.g., pooling method analysis and case study). In the unsupervised STS experiment and transfer tasks, the framework is initially fine-tuned on task-irrelevant large-scale NLI datasets [18], [19] and evaluated directly on classic STS datasets and transfer tasks datasets without additional training. Comparatively, in supervised experiments, we train and test CoSENT on datasets with various annotation types (e.g., positive-negative type, NLI type, and scoring type) to show the generality and effectiveness toward downstream tasks. Besides representation learning experiments, we extend the CoSENT to the interaction-based STS model and conduct comparative experiments. To demonstrate its adaptability towards different language models and pooling methods, we further investigate the CoSENT with Chinese pre-trained models and conduct an ablation study with different pooling methods. Moreover, we explore the convergence speed of CoSENT, which can be two to three times faster than SBERT. The experimental results achieve SOTA on the vast majority of datasets and tasks and provide strong evidence that our proposed framework can be consistently and efficiently optimized with different data annotations, for different pre-training and pooling methods, and on different STS tasks. All code and datasets of this work are publicly available at GitHub[1].

To summarize, the contributions of our work are as follows:

- An enlightening perspective is proposed to analyze the shortcomings of cosine similarity measurement in most BERT-based textual semantics, supported by empirical experiments.
- A novel, consistent, and adaptive framework based on similarity ranking, CoSENT, is presented for fine-tuning the transformer encoder to generate discriminative sentence embeddings.
- Extensive experiments are conducted on diverse datasets to confirm our method's new SOTA performance and excellent convergence speed.

The paper is organized as follows: Section II summarizes a series of representative works on sentence embedding and semantic textual similarity; Section III reviews the loss functions commonly used in STS tasks and describes the proposed framework, CoSENT; Section IV describes the training details of our framework, followed by the introduction of baselines and datasets used in the experiments. Section V shows rich experimental results on the basis of which the CoSENT's advantages are fully analyzed, and Conclusion (Section VI) discusses the implications and future work.

II. Related Work

A. Sentence Embedding Methods

Word embedding is the fundamental research in natural language processing, which aims to learn a unique representation vector for each word in the self-supervised manner [20], [21]. Early sentence embedding work extends this idea to more complex sentence representation models. Skipthought vector [22] and Sent2Vec [23] show that simply extending the skip-gram and n-gram models to sentence representation can lead to satisfactory results. [24] proposes a simple, unsupervised, neural network-free sentence embedding method and theoretically demonstrates its homogeneity with Word2Vec. Universal Sentence Encoder [25] first trains a transformer network to generate transferable sentence vectors.

Pre-trained language models like BERT [7], RoBERTa [26], and ELMo [27] take contextual embedding to a new level.

[1] https://github.com/RingBDStack/CoSENT

However, it is proven that the pre-trained models are incapable of directly generating semantically distinguishable sentence embeddings [8], as their induced embedding spaces are highly anisotropic. InferSent [11] introduces semantic similarity supervision between sentence pairs [18], [19] into the learning of sentence embeddings. Specifically, it extends the Siamese network architecture [28] with a variety of sophisticated encoders, including BiLSTM, the self-attentive network, and hierarchical ConvNet. Besides, it appends additional layers that perform concatenation, difference, and inner product before classification to capture correlated features between sentence representations. Sentence-BERT (SBERT) [12] verifies the effectiveness of fine-tuning the pre-trained BERT through the InferSent-like Siamese networks architecture. To accommodate different downstream tasks, SBERT employs three loss function schemes, namely cross-entropy loss, MSE loss, and triplet loss, for training, while uniformly calculating the cosine similarity between two sentence embeddings as the prediction. Meanwhile, a series of post-processing methods are proposed to address the anisotropy problem of pre-trained models. [9] empirically analyzes the poor performance of BERT in sentence representation and proposes BERT-flow, a post-processing approach based on normalizing flows to transform the distribution of BERT's embeddings into the Gaussian distribution. BERT-whitening [10] further demonstrates that a simple linear whitening operation can also effectively alleviate the anisotropy problem and can significantly reduce the dimensionality of the embeddings. Although SBERT effectively refines BERT sentence vectors, it introduces extra datasets with supervision. New studies attempt to achieve unsupervised training through data augmentation and contrastive learning. Analogous work [29] uses an additional frozen BERT encoder for similar samples, and ConSERT [30] explicitly introduces a data augmentation module to construct positive examples in various manners. SimCSE [31] proposes a simple contrastive learning framework for STS. Its supervised version exploits 3-tuple data (i.e., anchor sentence, similar sentence, and dissimilar sentence) constructed from the NLI dataset. Meanwhile, for each input sentence, the unsupervised version treats the augmentation obtained by applying different dropout masks as positive samples and the other inputs within the same mini-batch as negative samples. ArcCSE [32] improves SimCSE by adding an additive angular margin m between the positive pair and improves the triplet loss via modeling the entailment relation among triplet sentences. However, it still focuses on enhancing the contrastive and triplet loss used in SimCSE, which only aim to optimize the single sample with its positive and negative samples. DiffCSE [33] improves sentence embeddings by contrasting original sentences with their stochastically modified counterparts, generated through random masking and sampling from a masked language model. RankCSE [34] boosts performance by transferring ranking consistency information from the teacher to student models. It proposes the introduction of global ranking information into the model using ListNET and ListMLE losses, which are similar to our CoSENT loss. By employing CoSENT loss in the RankCSE model, we will make a complete comparison with it in Section V.

B. Semantic Textual Similarity Tasks

Semantic textual similarity (STS) is defined as the degree of semantic equivalence between two blocks of text. Measuring STS is a foundational technique in natural language understanding and is widely used in a variety of tasks, such as automated summarization [35], question answering [36], and text embedding [11]. Recent STS models can be categorized into two paradigms: representation-based and interaction-based. Representation-based STS models learn fixed representation vectors for sentences, with the semantic similarity between them defined by a simple binary function. Therefore, representation-based models are widely used in sentence embeddings [12] and time-sensitive tasks. Meanwhile, in the interaction-based scheme, sentence pairs are directly encoded to exploit cross-features or attentions between sentences, resulting in higher accuracy but also higher computational costs. A series of works such as BiMPM [37], ESIM [38], and DRCN [39] explicitly establish word-by-word interaction between sentence pairs based on recurrent neural network encoders. BERT [7] also provides a simple yet effective interaction-based text-matching scheme by concatenating two sentences and feeding them into the pre-trained BERT to produce a single vector, which is then used to calculate the similarity score.

III. A Consistent Sentence Embedding via Pairwise-similarity Ranking

This section will first review the loss functions used in STS tasks to explain our motivation. Then, we will present the formulation of the CoSENT loss function for the binary STS task and demonstrate how it can be generalized to datasets containing sortable labels and interactive models. Finally, we will describe the details of the architecture of our model.

A. Loss Function for STS Learning

Before introducing CoSENT, we first review loss functions used in STS tasks and sentence embedding to elaborate on our motivation. These include contrastive loss [40], MSE loss [41], softmax loss, and triplet loss [42].

Contrastive loss. [40] proposes a pairwise contrastive loss for the Siamese network, which is formulated as:

$$\mathcal{L}_{contrast} = \frac{1}{2} y_{true} d^2 + \frac{1}{2}(1 - y_{true}) \max(0, \epsilon - d)^2, \quad (1)$$

where d is the pair-wise distance, y_{true} is 1 if the pair is similar and 0 otherwise. ϵ is a predefined threshold representing the upper bound of the distance d. After replacing d with the cosine similarity $\cos(u_i, u_j)$ of the sentence embedding pair u_i and u_j to make it suitable for STS tasks, the contrastive loss can be modified as:

$$\begin{aligned}\mathcal{L}_{contrast} = &\frac{1}{2} y_{true}(1 - \cos(u_i, u_j))^2 \\ &+ \frac{1}{2}(1 - y_{true}) \max(\epsilon, \cos(u_i, u_j))^2.\end{aligned} \quad (2)$$

It drives the cosine similarity between positive pairs to converge to 1 and that between negative pairs to decrease to the threshold ϵ. However, in practical applications, selecting the optimal threshold ϵ for different datasets may be challenging. For instance, when dealing with datasets with a large proportion of hard samples, ϵ should not be set too low, as semantically dissimilar samples may only exhibit slight textual differences. For verification, we train SBERT on the five datasets mentioned in Section IV-D using the modified contrastive loss in Eq. 2 and report the results in Table I. It can be observed that for datasets with more hard samples (e.g., PAWS and ATEC), the selection of ϵ can significantly impact performance. InfoNCE [43] loss proposes to utilize the cross-entropy loss for contrastive learning as follows:

$$\mathcal{L}_{InfoNCE} = -\log \frac{e^{cos(u_i, u_i^*)/\tau}}{\sum_{j=1}^{n} e^{cos(u_i, u_j)/\tau}}, \tag{3}$$

where u_i^* is the positive sample of u_i, n is the batch size, and τ is a temperature hyperparameter. InfoNCE is widely used in SimCSE family models. It avoids directly defining the optimization threshold ϵ and replaces it with τ. However, the temperature is critical in controlling the local separation and global uniformity of the embedding distributions [44], so it needs to be adjusted for different datasets.

MSE loss. The MSE (Mean Squared Error) loss, also known as L2 loss, is a common loss function used in regression tasks. It is defined as the mean squared difference between the predicted outputs and the ground truth. In the STS task, the outputs are the similarities (e.g., cosine similarity) between the sentence pair (i, j):

$$\mathcal{L}_{MSE} = ||\cos(u_i, u_j) - y_{true}||^2, \tag{4}$$

where u_i and u_j are the embeddings of sentence pair i and j, while y_{true} is the similarity label. However, MSE loss is unsuitable for classification tasks because it turns all biases positive and amplifies outlier effects, making it more suitable for problems where noise in the observations follows a normal distribution [45].

Softmax loss. The softmax loss combines a softmax classifier and a cross-entropy loss and is widely used in classification tasks. In STS tasks, several different sentence embedding concatenation modes have been designed as input to a softmax classifier. InferSent [11] and Universal Sentence Encoder [25] both use $(u_i, u_j, |u_i - u_j|, u_i * u_j)$ as input, where u_i and u_j are the embeddings of the sentence pair i and j respectively in the Siamese network, while SBERT [12] uses $(u_i, u_j, |u_i - u_j|)$ as the input of the softmax classifier. The formal representation is written as follows:

$$\begin{aligned} h &= W(u_i, u_j, |u_i - u_j|), \\ \mathcal{L}_{softmax} &= -\log \frac{e^{h_{y_{true}}}}{\sum_{i=1}^{n} e^{h_i}}, \end{aligned} \tag{5}$$

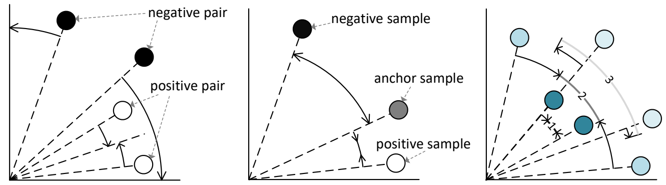
where W is a trainable matrix, h is the hidden layer output, and n is the number of classes in the multi-class classifier. h_i represents the i-th value in h and $h_{y_{true}}$ is the score of the target class. Although these models successfully use a softmax classifier to deviate the embedding vector from the initial anisotropic state, they fail to explain why cosine similarity is valid in the learned space. This gap in explanation arises because the cosine function is not involved in the training, yet it is used in STS prediction.

Triplet loss. Triplet loss [42] takes the triplet consisting of an anchor sentence embedding u_a, a positive sentence embedding u_p, and a negative sentence embedding u_n as the input. It is optimized to ensure that the distance between u_a and u_p is smaller than the distance between u_a and u_n in the learned space. It can be written as:

$$\mathcal{L}_{triplet} = max(d(u_a, u_p) - d(u_a, u_n) + \epsilon, 0), \tag{6}$$

where $d(u_i, u_j)$ is the distance of the sentence pair (i, j). The margin ϵ ensures that u_p is at least ϵ closer to u_a than to u_n.

The triplet loss function is sensitive to the choice of negative samples [46]. As models cannot learn anything from easy negative samples, hard negative samples are vital for enhancing the model's predictive accuracy. However, identifying and annotating a sufficient number of hard negative samples is challenging, making training with the triplet loss function both time-consuming and resource-intensive. Additionally, setting the margin ϵ poses its own challenges. If ϵ is too high, the model may struggle to learn the desired embedding space, while if ϵ is too low, the model risks overfitting the training data [47].

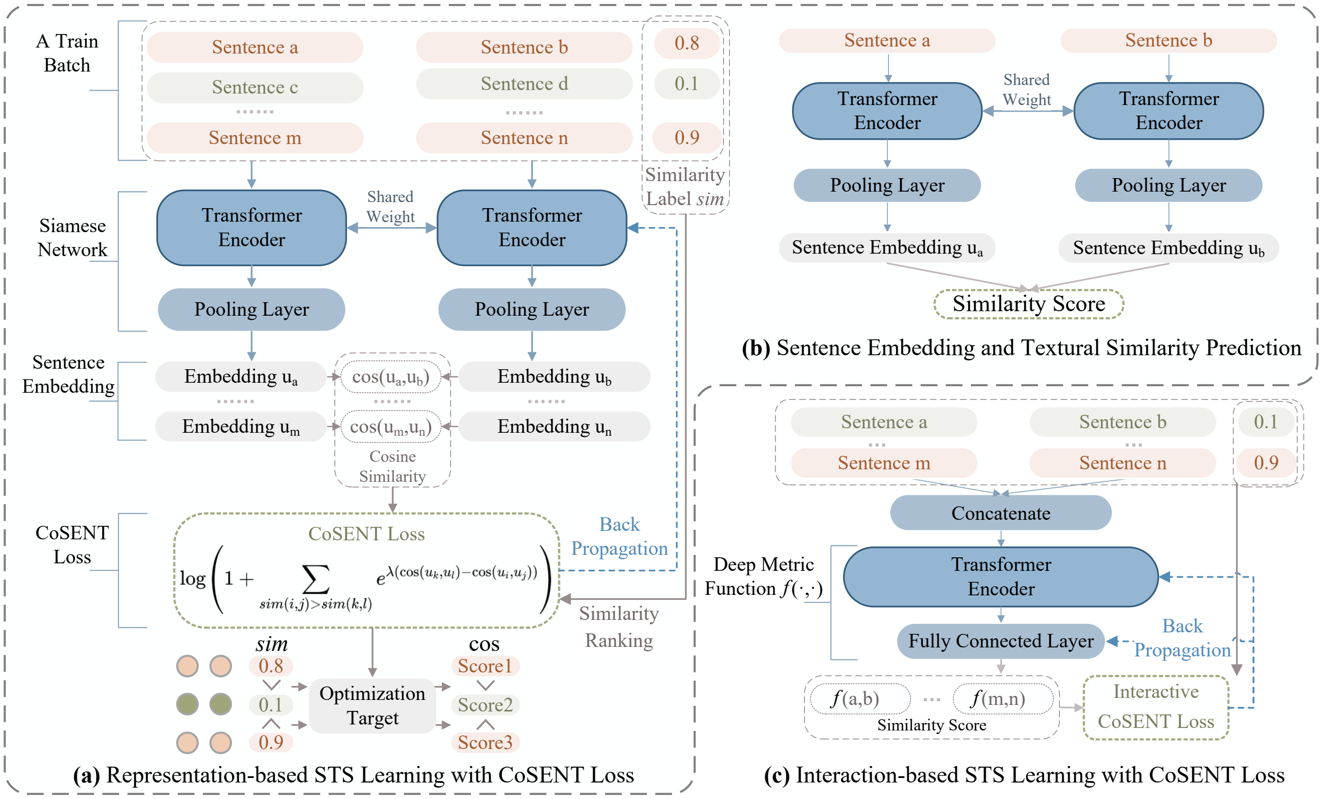


(a) Contrastive loss (b) Triplet loss (c) CoSENT loss

Fig. 1: Illustration of the optimization process of different loss functions. (a) shows the optimization direction of contrastive loss, while (b) shows the optimization using triplet loss. (c) illustrates the basic idea of CoSENT loss. The darker the node pairs (denoting sample pairs) in the figure, the higher their annotated similarity. By capturing the ranking of similarity annotations, the goal of CoSENT is to flexibly scale the distances between samples to obtain a representative sentence embedding space.

B. CoSENT Loss: General and Effective Optimization Objective

Due to the limitations of the above loss functions, we aim to design a loss function that directly optimizes the cosine function for consistency in training and prediction and can be widely used across various datasets. In this subsection, we introduce the core innovation of our model: a novel, similarity-ranking-based loss function for supervised STS training, named CoSENT. The comparison of common loss and CoSENT loss is illustrated in Fig. 1. The contrastive loss and triplet loss both focus on one sentence in relation to other single sentences, bringing positive samples closer and pushing negative samples further away, operating within the sentence pair. CoSENT, on the other hand, works on the sentence pairs and focuses on maintaining ranking consistency between the learned similarity of sentence pairs within the entire set and



(a) Representation-based STS Learning with CoSENT Loss

(b) Sentence Embedding and Textural Similarity Prediction

(c) Interaction-based STS Learning with CoSENT Loss

Fig. 2: The overall architecture of CoSENT.

their similarity labels. This approach enables the model to determine the appropriate similarity distribution.

1) CoSENT Loss for Binary Classification: Based on the idea presented in Section III-A, we consider that the distance between positive sample pairs is generally smaller than that between negative sample pairs. Formally, denoting Ω_{pos} and Ω_{neg} as the positive/negative sentence pairs set respectively, for any $(i, j) \in \Omega_{\text{pos}}$ and $(k, l) \in \Omega_{\text{neg}}$, it holds that:

$$\cos\left(u_i, u_j\right) > \cos\left(u_k, u_l\right), \tag{7}$$

where u_i , u_j, u_k, and u_l are the corresponding representations of sentences i, j , k, and l. The supervision focuses only on the similarity ranking between sample pairs with different labels, and the model itself determines the specific pairwise distance. Therefore, the binary CoSENT loss function can be written as:

$$\log\left(1 + \sum_{(i,j)\in\Omega_{\text{pos}},(k,l)\in\Omega_{\text{neg}}} e^{\lambda(\cos(u_k,u_l)-\cos(u_i,u_j))}\right), \tag{8}$$

where λ is a hyperparameter for amplification. It is derived from the cross-entropy loss with the softmax activation function:

$$-\log\frac{e^{s_t}}{\sum\limits_{i=1}^{n} e^{s_i}} = \log\left(1 + \sum_{i=1, i\neq t}^{n} e^{s_i - s_t}\right), \tag{9}$$

where n is the number of classes, $\{s_1, s_2, \dots, s_n\}$ represents the prediction score of each class, and s_t is the prediction score of the target class according to the supervision. It can be observed that its optimization objective is essentially to limit the scores of all non-target classes to be lower than the score of the target class, which is similar to ours.

2) Generalized CoSENT Loss for Ranked Labels: In the binary classification task, we expect the sentence vectors of the positive sample pairs to be more similar to each other than the sentence vectors of the negative sample pairs, which can be generalized to any rankable sample pair. Therefore, using sim to denote the similarity label of a sentence pair, we extend Eq. 9 as follows:

$$\log\left(1 + \sum_{sim(i,j)>sim(k,l)} e^{\lambda(\cos(u_k,u_l)-\cos(u_i,u_j))}\right). \tag{10}$$

As long as we can design the ranking of similarity for the sample pairs, we can use Eq. 10 to train the sentence vectors so that the higher the similarity between the sample pairs, the higher the cosine similarity of their sentence vectors will be. In practice, CoSENT can be adopted for various types of datasets with sortable labels, such as NLI datasets, positive-negative datasets, and scoring datasets.

3) CoSENT Loss for interaction-based STS Task: So far, our CoSENT loss functions have been designed to optimize the cosine similarity between sentence vectors, which is a representation-based model design. In representation-based models, the sentence pair is encoded separately using encoders, whereas interaction-based models treat the text pair as a single entity and classify it accordingly. CoSENT can also be used as a loss function for interaction-based models because it is a function that depends only on the relative ranking of the labels and is not necessarily related to cosine similarity. Denoting the output of the interaction model as $f(i,j)$ for any sentence pair (i,j), CoSENT loss for the interaction model is as follows:

$$\log\left(1+\sum_{sim(i,j)>sim(k,l)} e^{\lambda(f(k,l)-f(i,j))}\right). \tag{11}$$

C. Detailed Architecture

In this subsection, we will provide more details on the model. Fig. 2 (a) and (b) depict the training and prediction architecture of the representation-based model, respectively. During training, the model inputs all sentence pairs into the dual transformer pair by pair, and the output is passed through a pooling layer to obtain pairs of sentence embeddings. The pairs of sentence vectors are sorted according to their label, and the cosine similarities of the sentence vector pairs are added to the CoSENT loss based on the ranking information. The loss is then used to update the parameters of the Siamese network simultaneously through backpropagation, with the optimization target of maintaining ranking information consistency between the similarity labels and the predicted cosine similarity. Like SBERT, we use a Siamese network structure to update the transformer parameters, which enables the sentence vectors to share the same semantic space and be comparable using cosine similarity. At inference, the fine-tuned model computes the cosine similarity between sentence pairs as its output. The pooling layer after the Transformer encoder produces fixed-sized sentence embeddings. We experiment with four pooling strategies: Mean pooling takes the average of the hidden states produced by the transformer for each input token; CLS pooling uses the special CLS token; Max pooling involves taking the maximum value of the hidden states; and first-last pooling averages the first and the last tokens of the transformer.

In Fig. 2 (c), we present the interaction-based model. Sentence pairs are concatenated as the input of the transformer encoder. CoSENT then utilizes the output of the fully connected layer and corresponding labels to optimize the model. After training, the model directly uses the output of the fully connected layer as the similarity score for the STS task.

IV. Experimental Settings

A. Hardware and Software

We fine-tune the BERT$_{\text{large}}$ and RoBERTa$_{\text{large}}$ models on a server with an NVIDIA RTX A6000 GPU and an Intel i9-10980XE CPU. All other experiments are completed on a Linux server consisting of a 16-core Intel i9-12900 CPU, 32GB of RAM, and an NVIDIA GeForce RTX 3090Ti GPU. We implement all models using Pytorch 1.12 and Python 3.9.

B. Baselines

We compare the performance with the following baselines. Avg. GloVe embeddings denotes averaging all word embeddings generated by GloVe [20] to create sentence embeddings. Avg. BERT embeddings and BERT CLS-vector denote the average of the output or the CLS-token of raw BERT [7], respectively. InferSent-GloVe is the InferSent [11] model using the GloVe word embeddings as the input representation. Universal Sentence Encoder [25] is a transformer-based model designed to capture the relationships between different words in a sentence and generate a representation of the input text. SBERT-NLI and SRoBERTa-NLI correspond to the BERT and RoBERTa models fine-tuned on the NLI dataset using the SBERT [12] training approach. The terms cosine and softmax in parentheses indicate the loss function selected by the model. BERT$_{\text{base}}$-SimCSE refers to the unsupervised SimCSE [31] model that fine-tunes on the English Wikipedia dataset using dropout as a data augmentation strategy. ArcCSE [32] enhances the pairwise discriminative power and models the entailment relation among triplet sentences. DiffCSE [33] introduces equivariant contrastive learning to SimCSE. RankCSE [34] utilizes a teacher-student framework to learn additional ranking information. In the SentEval transfer task, BERT$_{\text{base}}$-SimCSE-sup is the supervised SimCSE [31] model trained on the NLI dataset. As supervised results are not reported for ArcCSE, DiffCSE, and RankCSE, we only compare with the unsupervised versions of these models.

C. Training Details

In experiments, we adopt a consistent set of hyperparameters, including a learning rate of 2e-5 and a weight decay of 0.01. A linear learning rate warm-up is applied over the first 10% of the training data. For fine-tuning on the NLI dataset in unsupervised tasks, we train the models for one epoch with a batch size of 64. In other situations, the default number of training epochs is four and the batch size is 16. For unsupervised tasks, the default pooling method for BERT is the first-last pooling, whereas the CLS pooling method is utilized in the RoBERTa model. The mean pooling method is selected for other tasks. We set the hyperparameter λ in Eq. 10 to a uniform value of 20 for all datasets. BERT$_{\text{base}}$-RankCSE-CoSENT is trained using the original experimental settings of RankCSE [34] and employs our CoSENT loss to capture the consistent ranking information between the teacher and student models. In the English task, BERT$_{\text{base}}$ and BERT$_{\text{large}}$ refer to BERT-base-uncased and BERT-large-uncased models presented in [7], while RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$ refer to RoBERTa-base and RoBERTa-large models proposed in [26]. In the Chinese task, BERT$_{\text{base}}$ refers to bert-base-chinese [7] and RoBERTa$_{\text{base}}$ refers to chinese-roberta-wwm-ext [48]. All models are downloaded from Huggingface.

D. Dataset

We employ a range of English and Chinese STS datasets, including the Wiki [31], STS 2012-2016 [49], [50], [51], [52], [53], STS benchmark [14], SICK [15], and PAWS [13] datasets for the English task, and the ATEC [16], BQ [17], LCQMC [54], PAWSX [55], and Chinese-STSb [56] datasets for the Chinese task. These datasets are described in detail as follows.

TABLE II: Statistics of benchmark datasets.

Dataset	Train	Validation	Test	Similarity label
Wiki	1000000	-	-	-
NLI	942069	19657	19656	0, 1, 2
STS 2012	2234	6	3108	0-5
STS 2013	-	-	1500	0-5
STS 2014	-	-	3750	0-5
STS 2015	-	-	3000	0-5
STS 2016	-	-	1186	0-5
STS benchmark	5749	1500	1379	0-5
SICK	4439	495	4906	1-5 / 0, 1, 2
PAWS	49401	8000	8000	0, 1
ATEC	62477	20000	20000	0, 1
BQ	100000	10000	10000	0, 1
LCQMC	238766	8802	12500	0, 1
PAWSX	49401	2000	2000	0, 1
Chinese-STSb	5231	1458	1361	0-5

Wiki. The Wiki dataset is provided by SimCSE, which contains 10^6 randomly sampled sentences from English Wikipedia.

STS 2012-1016. The STS 2012-2016 datasets are built by the organizer of the SemEval shared task. Each sample in the dataset consists of a pair of sentences, along with a label indicating the similarity of the two sentences on a scale from 0 to 5. A score of 0 signifies that the semantics of the sentences are completely independent, while a score of 5 indicates that the sentences are semantically equivalent.

STS benchmark. The STS benchmark dataset consists of a carefully chosen group of English datasets used in the STS tasks organized in the context of SemEval between 2012 and 2017. The datasets encompass text sources ranging from image captions, and news headlines to user forums. The labels of this dataset are inherited from the STS 2012-2016 datasets.

NLI. The NLI (Natural Language Inference) dataset is a combination of the SNLI [18] and the Multi-Genre NLI [19] dataset. The label of the NLI dataset contains the categories of contradiction, neutral, and entailment.

SICK The SICK (Sentences Involving Compositional Knowledge) dataset consists of 10,000 sentence pairs from two existing sets: the 8K ImageFlickr dataset and the SemEval 2012 STS MSR-Video Description dataset. Each sentence pair has a relatedness score and a text entailment relation. The dataset can be referred to as the SICK-Relatedness dataset when the relatedness score (1-5) is considered and as the SICK-NLI dataset when the NLI label (contradiction, neutral, or entailment) is applied.

PAWS. The PAWS dataset contains 108,463 pairs that have been labeled by humans and an additional 656,000 pairs that have been labeled with noise. We only use a subset of the PAWS dataset known as the PAWS-Wiki labels (Final) dataset, which includes pairs generated through word swapping and back translation methods. These pairs have received human judgments regarding both paraphrasing and fluency. The labels for each sentence pair in the dataset are binary, with 0 indicating that the pair has a distinct meaning, and 1 indicating that the pair is a paraphrase.

ATEC. The ATEC dataset is derived from sentence pairs of actual customer service interactions in Ant Financial's Brain application.

BQ. The BQ (Bank Question) dataset is a large-scale, domain-specific Chinese corpus containing 120,000 question pairs from online bank customer service logs.

LCQMC. The LCQMC dataset uses Baidu Knows as the original data source to collect large-scale sentence pairs.

PAWSX. The PAWSX dataset contains 23,659 human-translated PAWS evaluation pairs and 296,406 machine-translated training pairs in six typologically distinct languages. We only use the Chinese subset.

Chinese-STSb. The Chinese-STSb dataset is obtained by translating the raw English STS benchmark dataset using Tencent Cloud's API and then undergoing manual revision to address errors and inaccuracies in the sentences.

V. Experimental Results and Analysis

To demonstrate the effectiveness of our proposed method, we present our experimental results for various tasks related to semantic textual similarity in this section. As the Pearson correlation is proven inadequate for STS tasks [57], we uniformly report Spearman's rank correlation between the cosine similarity and the gold labels to compare the performance of the models. The reported results of the baselines are partially derived from the original paper and partially evaluated using the authors' source code.

A. Unsupervised STS

In the unsupervised STS task, we only fine-tune models on the NLI dataset or the Wikipedia dataset, not using any STS-specific training data. Then we evaluate model performance on the STS 2012-2016 dataset [49], [50], [51], [52], [53], the STS benchmark dataset [14], and the SICK-Relatedness dataset [15]. The results are reported in Table III. It shows that the raw BERT sentence embeddings, with both the CLS-token and averaging pooling methods, fail to outperform the averaged GloVe embeddings in all datasets. For the $\mathrm{BERT}_{\mathrm{base}}$, $\mathrm{BERT}_{\mathrm{large}}$, $\mathrm{RoBERTa}_{\mathrm{base}}$, and $\mathrm{RoBERTa}_{\mathrm{large}}$ models, using the CoSENT loss to fine-tune on the NLI dataset generally results in better performance on the 7 datasets compared to the other baselines, with a maximum average improvement of 1.13 on the datasets compared to SBERT. Compared with other SimCSE family models fine-tuned on the Wiki dataset, $\mathrm{BERT}_{\mathrm{base}}$-RankCSE-CoSENT achieves an average improvement ranging from 0.55 to 4.66 in Spearman's rank correlation across all datasets. Moreover, $\mathrm{BERT}_{\mathrm{base}}$-RankCSE-CoSENT outperforms RankCSE with LitNet and ListMLE loss in 5 out of 6 datasets, which demonstrates the benefit of our CoSENT loss in ranking consistency information learning.

We also observe that the RoBERTa model generally outperforms the BERT model. However, the performance enhancement of CoSENT with BERT is more prominent than those with RoBERTa. Specifically, CoSENT improves $\mathrm{BERT}_{\mathrm{base}}$ and $\mathrm{BERT}_{\mathrm{large}}$ on six and seven datasets, respectively, while only improving $\mathrm{RoBERTa}_{\mathrm{base}}$ and $\mathrm{RoBERTa}_{\mathrm{large}}$ on four and three datasets, respectively. We analyze that this is because the raw RoBERTa model is more sufficient compared to the BERT model.

TABLE III: Results of unsupervised STS tasks. We report Spearman's rank correlation ρ between the cosine similarity of sentence representations and the gold labels for various datasets. Performance is reported as $\rho \times 100$. STS12-STS16 refers to the STS 2012-2016 datasets, STSb refers to the STS benchmark dataset, and SICK-R refers to the SICK-Relatedness dataset. **Bold**: the best performance under each category, underline: the second best performance.

Model	Fine-tune data	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg.
Avg. GloVe embeddings	N/A	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. BERT embeddings	N/A	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT CLS-vector	N/A	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
InferSent - GloVe	NLI	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder	NLI	64.49	67.80	64.61	76.83	73.18	74.92	**76.69**	71.22
SBERT$_{\text{base}}$-NLI	NLI	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT$_{\text{large}}$-NLI	NLI	72.27	78.46	74.90	80.99	76.25	79.23	73.75	76.55
SRoBERTa$_{\text{base}}$-NLI	NLI	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa$_{\text{large}}$-NLI	NLI	**74.53**	77.00	73.18	**81.85**	76.82	79.10	74.29	76.68
BERT$_{\text{base}}$-CoSENT-NLI	NLI	71.34	76.06	73.63	80.71	75.23	78.25	74.10	75.62
BERT$_{\text{large}}$-CoSENT-NLI	NLI	74.44	**79.17**	**76.10**	81.13	**77.88**	**80.35**	74.68	**77.68**
RoBERTa$_{\text{base}}$-CoSENT-NLI	NLI	72.29	76.79	74.31	78.64	76.29	77.75	68.70	74.97
RoBERTa$_{\text{large}}$-CoSENT-NLI	NLI	74.31	77.87	75.36	79.63	76.38	79.11	69.56	76.03
BERT$_{\text{base}}$-SimCSE	Wiki	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
BERT$_{\text{base}}$-ArcCSE	Wiki	72.08	84.27	76.25	82.32	79.54	79.92	72.39	78.11
BERT$_{\text{base}}$-DiffCSE	Wiki	72.28	84.43	76.47	83.90	80.54	80.59	71.23	78.49
BERT$_{\text{base}}$-RankCSE-listNet	Wiki	74.38	85.97	77.51	84.46	81.31	81.46	75.26	80.05
BERT$_{\text{base}}$-RankCSE-listMLE	Wiki	75.66	**86.27**	77.81	**84.74**	81.10	81.80	75.13	80.36
BERT$_{\text{base}}$-RankCSE-CoSENT	Wiki	**75.76**	85.54	**78.11**	84.60	**81.58**	**82.88**	**77.92**	**80.91**

TABLE IV: Results of supervised STS tasks. The training and testing data come from distinct divisions of the same dataset. We report Spearman's rank correlation ρ between the cosine similarity of sentence representations and the gold labels on multiple datasets. **Bold**: the best performance under each category, underline: the second best performance, "–": results are not available.

Model	NLI	STSb	SICK-R	PAWS	MRPC
SBERT$_{\text{base}}$(MSE)	76.60	84.67	83.76	73.09	55.46
SBERT$_{\text{large}}$(MSE)	76.97	84.45	84.99	74.15	53.49
SRoBERTa$_{\text{base}}$(MSE)	77.80	84.92	84.47	72.57	61.55
SRoBERTa$_{\text{large}}$(MSE)	74.01	85.02	85.12	74.11	60.46
SBERT$_{\text{base}}$(softmax)	55.52	-	-	71.96	41.06
SBERT$_{\text{large}}$(softmax)	57.26	-	-	63.71	41.02
SRoBERTa$_{\text{base}}$(softmax)	56.61	-	-	73.46	42.11
SRoBERTa$_{\text{large}}$(softmax)	60.84	-	-	70.01	44.76
BERT$_{\text{base}}$-CoSENT	77.88	85.75	84.43	74.59	62.16
BERT$_{\text{large}}$-CoSENT	77.01	86.40	85.00	**76.26**	60.17
RoBERTa$_{\text{base}}$-CoSENT	79.12	86.95	84.64	75.81	**67.31**
RoBERTa$_{\text{large}}$-CoSENT	**87.04**	**87.84**	**85.40**	75.63	66.13

B. Supervised STS

In the supervised STS task, we directly fine-tune the model on the training subset of the NLI [18], [19] dataset, STS benchmark dataset [14], SICK-Relatedness dataset [15], PAWS dataset [13], and MRPC dataset, and evaluate it on the corresponding test partition of each dataset. Since the STSb and SICK-R datasets are scoring datasets, the softmax loss is not available. Results are reported in Table IV. It demonstrates the efficiency of the CoSENT over all other loss functions used in SBERT across datasets and pre-trained models. Concretely, CoSENT consistently outperforms its competitors with a maximum average improvement ranging from 0.04 to 26.20 in the NLI dataset, from 1.08 to 2.82 in the STSb dataset, from 0.01 to 0.67 in the SICK-R dataset, from 1.50 to 12.55 in the PAWS dataset, and from 5.67 to 25.20 in the MRPC dataset.

It is noteworthy that the performance of the softmax loss in the supervised STS task is subpar in comparison to other loss functions. We suggest that this underperformance is due to the inconsistency between the softmax loss optimization objective and Spearman's rank coefficient of cosine similarity used in the prediction phase. Interestingly, the softmax loss produces relatively satisfactory sentence embeddings in the unsupervised STS task.

C. Chinese STS

Considering the differences in semantic space complexity and corresponding pre-trained model capabilities of different languages, we also conduct experiments on various Chinese datasets and on the basis of Chinese pre-trained models in addition to the classical STS benchmarks.

We evaluate both supervised and unsupervised STS tasks using five datasets: ATEC[16], BQ[17], LCQMC[54], PAWSX[55], and Chinese-STSb [56] dataset. All results are reported in Table VI. Our observations can be summarized as follows: **(1)** In the unsupervised STS tasks, CoSENT achieves general improvement across five datasets, especially on the PAWSX dataset, with an average improvement of up to 5.25. This demonstrates that the sentence embedding space learned by CoSENT is superior to that learned by SBERT in the Chinese task. **(2)** In the supervised STS tasks, CoSENT achieves improvement across all datasets. The improvement is quantified with a maximum increase of 5.86 and 2.17 when compared to the best results obtained from softmax and MSE methods in the BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ models. **(3)** MSE loss is not suitable for binary classification datasets, such as ATEC and LCQMC, while the softmax loss is not available for the scoring dataset STSb. In comparison, CoSENT can be adopted in various types of datasets and achieves superior performance.

D. Transfer Task - SentEval

SentEval [58] is a library for evaluating the quality of sentence embeddings. It provides a set of predefined transfer tasks that can be used to evaluate the performance of sentence embeddings. The goal of the task is to train a classifier to

TABLE V: Evaluation of sentence embeddings on transfer tasks using the SentEval toolkit. SentEval evaluates sentence embeddings on different sentence classification tasks by training a logistic regression classifier using the sentence embeddings as features. Classification accuracy (%) based on 10-fold cross-validation is reported. **Bold**: the best performance under each category, underline: the second best performance.

Model	Fine-tune data	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Avg. GloVe embeddings	N/A	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Avg. fast-text embeddings	N/A	77.96	79.23	91.68	87.81	82.15	83.60	74.49	82.42
Avg. BERT embeddings	N/A	78.66	86.25	94.37	88.66	84.40	92.80	69.45	84.94
BERT CLS-vector	N/A	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
InferSent - GloVe	NLI	81.57	86.54	92.50	90.38	84.18	88.20	75.77	85.59
Universal Sentence Encoder	NLI	80.09	85.19	93.98	86.70	86.38	**93.20**	70.14	85.10
SBERT$_{base}$-NLI	NLI	83.64	89.43	94.39	89.86	88.96	89.60	76.00	87.41
SBERT$_{large}$-NLI	NLI	84.88	90.07	94.52	90.33	90.66	87.40	75.94	87.69
BERT$_{base}$-SimCSE	Wiki	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
BERT$_{base}$-SimCSE-sup	NLI	82.69	89.25	94.81	89.59	87.31	88.40	73.51	86.51
BERT$_{base}$-ArcCSE	Wiki	79.91	85.25	**99.58**	89.21	84.90	89.20	74.78	86.12
BERT$_{base}$-DiffCSE	Wiki	81.76	86.20	94.76	89.21	86.00	87.60	75.54	85.87
BERT$_{base}$-RankCSE-listNet	Wiki	83.21	88.08	95.25	90.00	88.58	90.00	76.17	87.33
BERT$_{base}$-RankCSE-listMLE	Wiki	83.07	88.27	95.06	89.90	87.70	89.40	76.23	87.09
BERT$_{base}$-RankCSE-CoSENT	Wiki	83.46	89.54	94.84	90.32	88.08	86.80	**77.45**	87.21
BERT$_{base}$-CoSENT-NLI	NLI	83.70	89.86	93.94	90.17	89.29	91.60	76.35	87.84
BERT$_{large}$-CoSENT-NLI	NLI	**85.31**	**90.44**	94.68	**90.52**	**90.94**	90.80	77.10	**88.54**

TABLE VI: Results of different Chinese STS datasets. The PAWSX dataset is a Chinese subset of the original PAWSX [55] dataset. STSb refers to the Chinese-STSb [56] dataset. The top half of the table shows the unsupervised STS results, and the bottom half shows the supervised STS results. **Bold**: the best performance under each category, underline: the second best performance, "–": results are not available.

Model	ATEC	BQ	LCQMC	PAWSX	STSb
SBERT$_{base}$-NLI	28.19	42.73	64.98	15.38	**74.88**
SRoBERTa$_{base}$-NLI	**31.87**	45.60	67.89	15.64	73.93
BERT$_{base}$-CoSENT-NLI	28.93	41.84	66.07	20.49	73.91
RoBERTa$_{base}$-CoSENT-NLI	31.84	**46.65**	**68.43**	**20.89**	74.37
SBERT$_{base}$(MSE)	45.55	70.74	77.83	51.76	76.29
SRoBERTa$_{base}$(MSE)	46.09	72.14	77.99	58.41	78.97
SBERT$_{base}$(softmax)	48.68	70.97	79.16	51.24	-
SRoBERTa$_{base}$(softmax)	50.15	71.06	79.48	59.71	-
BERT$_{base}$-CoSENT	50.13	71.53	79.45	57.62	**81.26**
RoBERTa$_{base}$-CoSENT	**50.84**	**72.60**	**79.89**	**61.63**	81.14

predict the label of a given sentence pair, using sentence embeddings as input. The accuracy of the classifier is used as a measure of the quality of the sentence embeddings. We compare sentence embeddings with baselines on the following seven transfer tasks.

MR [59]. MR involves sentiment classification of movie reviews. The goal is to predict whether a given movie review is positive or negative.

CR [60]. CR is similar to MR but it uses customer reviews from various products on e-commerce sites.

SUBJ [61]. The subjectivity dataset with subjective reviews and objective plot summaries. This task focuses on determining the subjectivity of a sentence (objective or subjective).

MPQA [62]. Phrase level opinion polarity classification task (positive, negative, neutral) on the MPQA Opinion Corpus.

SST [63]. The Stanford Sentiment Treebank (SST) transfer task aims to predict the sentiment of a given sentence on a five-point scale (very negative, negative, neutral, positive, very positive).

TREC [64]. The fine-grained question-type classification task, where the goal is to predict the type of question based on its text, such as factoid, list, yes/no, etc.

MRPC [65]. The Microsoft Research Paraphrase Corpus from parallel news sources. This task is a paraphrase identification task, which aims to determine whether two sentences are semantically equivalent.

The results are reported in Table V. CoSENT achieves the best performance in 5 out of 7 tasks. Compared to InferSent and Universal Sentence Encoder, CoSENT has an average increase of approximately 3% in performance. Additionally, it has an average increase of around 1% when compared to SBERT and an increase of approximately 2% when compared to SimCSE. CoSENT demonstrates a general improvement compared to SBERT, with the exception of the SST and MRPC tasks. This highlights the advantage of the quality of sentence embeddings learned by CoSENT. Compared with RankCSE, BERT$_{base}$-RankCSE-CoSENT outperforms in 4 out of 7 tasks, which demonstrates the advantage of CoSENT in capturing ranking information compared to the ListNet and ListMLE losses used in the RankCSE model. Additionally, CoSENT shows significant underperformance on the TREC dataset. We analyze this because TREC is a question-type classification task, which has a more specific sentiment space.

TABLE VII: Results of the interaction-based model. The suffix CE refers to the model with cross-entropy as the loss function. **Bold**: the best performance under each category.

Model	ATEC	BQ	LCQMC	PAWSX	Avg.
BERT$_{base}$-CE	50.35	73.66	79.33	**63.23**	66.64
BERT$_{base}$-CoSENT	**50.42**	**74.07**	**79.68**	63.07	**66.81**
RoBERTa$_{base}$-CE	50.32	72.49	**80.13**	70.33	68.31
RoBERTa$_{base}$-CoSENT	**50.54**	**73.27**	80.03	**70.35**	**68.55**

E. Interaction-based STS

We propose the CoSENT formula, as outlined in Eq. 11, as the loss function for the interaction-based model. In this subsection, we evaluate the performance of CoSENT in the interaction-based STS task. We fine-tune the raw BERT$_{base}$ and RoBERTa$_{base}$ models on the ATEC, BQ, LCQMC, and PAWSX datasets, and report the Spearman's rank correlation in Table VII. CoSENT slightly outperforms cross-entropy loss,

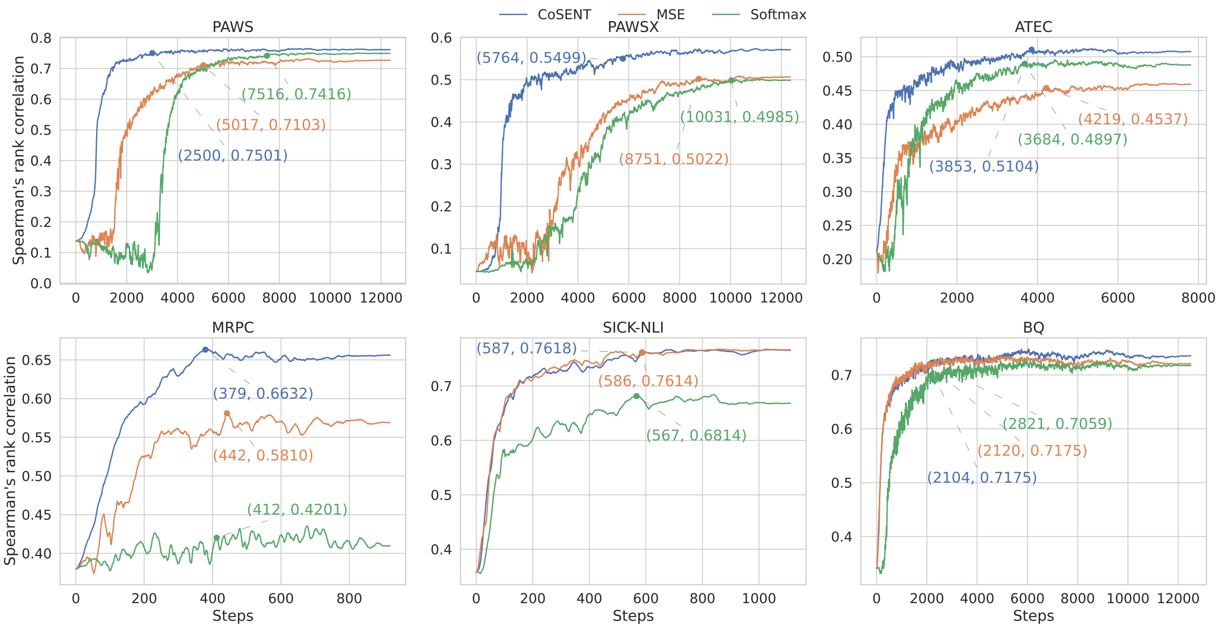


Fig. 3: The average convergence speed for three different loss functions: CoSENT, MSE, and Softmax, across various datasets. The x-axis represents the number of steps and the y-axis represents Spearman's rank correlation. The results indicate that the use of the CoSENT loss function can significantly reduce training time while also improving performance.

with an average improvement of 0.31 and 0.24 for BERT and RoBERTa, respectively. Furthermore, we observe that the performance of the BERT model is improved more by CoSENT compared to the RoBERTa model. This may be attributed to the fact that RoBERTa is a more robust and optimized version of BERT.

F. Experiment on Model Convergence Speed

In this experiment, we evaluate the convergence speed of CoSENT compared to other models. We use the same dataset and training setup for all models and measure both the training time and the final performance of each model.

As illustrated in Fig. 3, CoSENT demonstrates an improvement in convergence speed while also increasing model performance. Specifically, for datasets with hard samples like PAWS and PAWSX, fine-tuning with CoSENT can approximately reduce the training time by 50%-67% compared to the baseline while achieving the same level of performance as measured by Spearman's rank correlation. On the PAWS dataset, CoSENT achieves convergence at step 2500 with a value of 0.7501, while MSE loss converges at step 5017 with a value of 0.7103, and softmax loss converges at step 7516 with a value of 0.7416. Furthermore, CoSENT consistently outperforms other baselines throughout the entire training phase.

G. Ablation Study

In this subsection, we conduct an ablation study to gain a deeper understanding of the various components of CoSENT and their relative significance.

1) Hyperparameter Sensitivity: CoSENT has a single hyperparameter, denoted as λ, as shown in Eq. 10. To evaluate its impact, experiments are conducted by varying the value of λ from 1 to 40 in increments of 5, using both the raw transformer BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ models, across three diverse datasets. As shown in Fig. 4, the hyperparameter λ has little effect on the performance of the model, as the Spearman's rank correlation is similar when it is selected within the range of 5-40, except for RoBERTa$_{\text{large}}$ with a λ of 40. Therefore, we can confidently set the hyperparameter for different tasks without worrying about how the model performance will be affected.

TABLE VIII: Results of CoSENT with different pooling methods in supervised tasks. BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ are selected as the raw models for fine-tuning. **Bold**: the best performance under each category, underline: the second best performance.

Pooling Strategy	NLI	STSb	SICK-R	PAWS	MRPC	Avg.
BERT$_{\text{base}}$+Mean	**77.88**	**85.75**	**84.43**	74.59	**62.16**	**76.96**
BERT$_{\text{base}}$+CLS	75.80	85.73	83.97	**74.87**	59.95	76.06
BERT$_{\text{base}}$+Max	71.51	84.61	83.33	73.53	57.41	74.08
BERT$_{\text{base}}$+first-last	68.23	85.12	83.68	74.07	60.06	74.23
RoBERTa$_{\text{base}}$+Mean	**79.12**	**86.95**	84.64	**75.81**	**67.31**	**78.77**
RoBERTa$_{\text{base}}$+CLS	76.74	86.26	84.68	74.99	64.51	77.44
RoBERTa$_{\text{base}}$+Max	71.78	86.26	83.87	74.76	62.98	75.93
RoBERTa$_{\text{base}}$+first-last	74.40	86.73	**84.66**	74.85	66.17	77.36

2) Pooling Methods: Different pooling methods can be adopted in the pooling layer of the framework shown in Fig. 2. In this subsection, we evaluate various pooling strategies such as Mean, CLS, Max, and first-last, and report the results in Table VIII. We find that the impact of the pooling strategy on the STSb, SICK-R, and PAWS datasets is relatively minor, while it has a greater impact on the NLI and MRPC datasets. Additionally, we observe that the mean pooling strategy generally outperforms the other three pooling strategies on all datasets and models, with the exception of the SICK-R dataset when using the RoBERTa$_{\text{base}}$ model, where the mean pooling

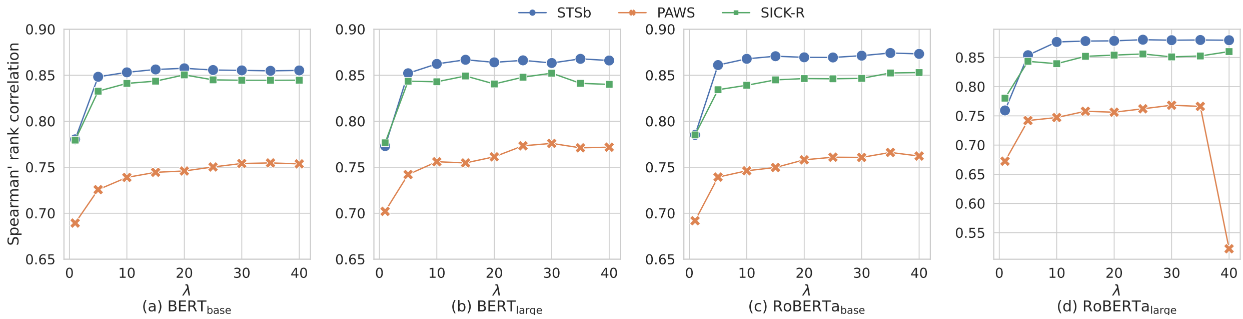


Fig. 4: The influence of varying hyperparameter λ values selected within CoSENT. Our experiment is conducted on three English STS datasets (STSb [14], PAWS [13], and SICK-R [15]) using both the BERT and RoBERTa models. The x-axis represents the value of hyperparameter λ and the y-axis represents Spearman's rank correlation.

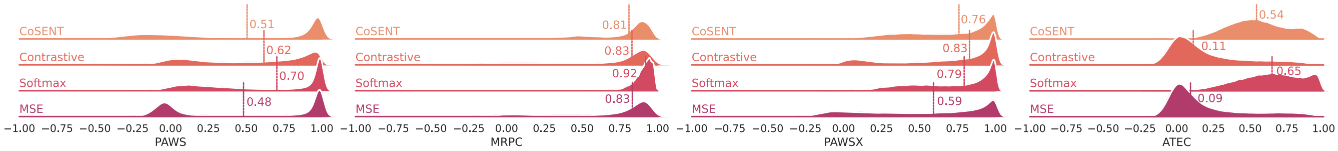


Fig. 5: Density plots of cosine similarities between sentence pairs. The x-axis is the cosine similarity.

TABLE IX: Results of CoSENT with different pooling methods in the unsupervised tasks. $\mathrm{BERT_{base}}$-NLI and $\mathrm{RoBERTa_{base}}$-NLI refer to the raw model fine-tuned on the NLI dataset. **Bold**: the best performance under each category, underline: the second best performance.

Pooling Strategy	STSb	SICK-R	PAWS	MRPC	Avg.
$\mathrm{BERT_{base}}$-NLI+Mean	76.22	69.79	29.19	50.21	56.35
$\mathrm{BERT_{base}}$-NLI+CLS	76.12	68.90	29.58	**50.41**	56.25
$\mathrm{BERT_{base}}$-NLI+Max	74.87	69.70	**35.42**	50.28	**57.57**
$\mathrm{BERT_{base}}$-NLI+first-last	**78.25**	**74.10**	19.43	50.37	55.54
$\mathrm{RoBERTa_{base}}$-NLI+Mean	76.54	68.51	34.24	53.04	58.08
$\mathrm{RoBERTa_{base}}$-NLI+CLS	**77.75**	68.70	32.70	52.85	58.00
$\mathrm{RoBERTa_{base}}$-NLI+Max	75.77	68.29	**35.28**	53.07	**58.10**
$\mathrm{RoBERTa_{base}}$-NLI+first-last	75.44	**69.85**	29.40	**53.83**	57.12

strategy slightly underperforms the CLS and first-last pooling strategies. In contrast, the max pooling strategy underperforms all other pooling methods on all datasets and models, except when using $\mathrm{BERT_{base}}$ and first-last on the NLI dataset, which is similar to the results found by Reimers et al [12].

We also conduct experiments to evaluate the impact of different pooling strategies on the unsupervised task. As shown in Table IX, the max pooling demonstrates an average improvement of 0.02 to 2.03 in comparison to other strategies, while it performs the worst on the supervised tasks. The pooling strategy has a significant impact on the PAWS dataset. This is because the PAWS dataset contains a large number of hard samples, which have high requirements for the learned sentence embedding space. Furthermore, we find that the first-last pooling strategy outperforms all other strategies on the STSb and SICK-R datasets for the $\mathrm{BERT_{base}}$-NLI models. This is in contrast to the situation in the supervised task, where first-last achieves the worst performance for the NLI dataset.

H. Cosine-similarity Distribution

To intuitively demonstrate the capability of CoSENT, we illustrate the distribution of cosine similarity for the fine-tuned BERT model using varying loss functions on the PAWS, MRPC, PAWSX, and ATEC datasets in Fig. 5. We consistently maintain a threshold ϵ of 0 for contrastive loss during all evaluations. The PAWS, PAWSX, and ATEC datasets contain numerous difficult negative samples, making it challenging for the contrastive loss to differentiate them if the threshold is not set to an appropriate level. Compared to the other loss, CoSENT typically exhibits a more dispersed distribution, leading to better performance on STS tasks. On the ATEC dataset, contrastive loss and MSE loss tend to concentrate around 0, as they drive the learning of the similarity of negative sample pairs towards 0. This may not be suitable for handling challenging samples.

I. Case Study

As a framework for learning sentence embeddings, CoSENT can be applied to a wide range of natural language processing tasks, such as sentence pair classification tasks, information retrieval, and machine translation. To showcase the versatility and effectiveness of CoSENT, we present a detailed case study and report the results in Table X. We select SBERT as the baseline and the raw $\mathrm{BERT_{base}}$ model for fine-tuning. We select six English sentence pairs and four Chinese sentence pairs from the PAWS and PAWSX datasets, respectively.

Both models succeed in sentence pairs 1 and 3, which only have a slight difference in word order and phrasing. In contrast, they fail in sentence 2. They judge "the Otago region of New Zealand" to be different from "the New Zealand region of Otago" but the difference in word order only conveys a subtle difference in emphasis and doesn't impact the conveyed information. The former places the focus on the specific region, Otago, within New Zealand, while the second focuses on the country, New Zealand, and the specific region, Otago, within it. The SBERT model struggles to accurately distinguish sentence pairs 4 and 5, where the subject changes

TABLE X: Case Study on both English and Chinese datasets. For each sample sentence pair, a comparison performance on the ground truth and the prediction of the baseline model and CoSENT is presented. Predictions in red indicate incorrect assignments, while those in green indicate correct ones.

Sentence pair 1	The NBA season of 1975 – 76 was the 30th season of the National Basketball Association.
	The 1975 – 76 season of the National Basketball Association was the 30th season of the NBA.
Ground Truth: Equivalent baseline: Equivalent ours:Equivalent	
Sentence pair 2	Taieri is a former parliamentary electorate in the Otago region of New Zealand, from 1866 to 1911.
	Taieri is a former parliamentary electorate in the New Zealand region of Otago, from 1866 to 1911.
Ground Truth: Equivalent baseline: Unequivalent ours:Unequivalent	
Sentence pair 3	The film was a commercial hit, and one of Sergio Sollima's more successful films, and less political than the director's earlier Spaghetti Westerns.
	The film was a commercial hit, and one of Sergio Sollima's more political films, and less successful than the former spaghetti-director's westerns.
Ground Truth: Unequivalent baseline: Unequivalent ours:Unequivalent	
Sentence pair 4	In 1900, Elizabeth married Waller Cowles, and her daughter Harriet was born in 1912.
	Elizabeth Waller married Cowles in 1900, and their daughter Harriet was born in 1912.
Ground Truth: Unequivalent baseline: Equivalent ours:Unequivalent	
Sentence pair 5	Abdul Rahman said Raouf will only survive if he goes into exile.
	Abdul Rahman will survive only if he goes into exile.
Ground Truth: Unequivalent baseline: Equivalent ours: Unequivalent	
Sentence pair 6	On July 30, 2012, it was announced that Corrêa should have a test with Middlesbrough FC after being recommended by Club legend Juninho Paulista to manager Tony Mowbray.
	On 30 July 2012 it was announced Corrêa would have a trial with Middlesbrough FC after being recommended to manager Tony Mowbray by club legend Juninho Paulista.
Ground Truth: Equivalent baseline: Unquivalent ours: Equivalent	
Sentence pair 7	还有具体的讨论，公众形象辩论和项目讨论。
	还有公开讨论，特定档案讨论和项目讨论。
Ground Truth: Unequivalent baseline: Unequivalent ours: Unequivalent	
Sentence pair 8	现有长期教师43 人，其中大多数教师是国家优秀艺术家。
	有43 名常任教师，而且大多数教师都是该国著名的艺术家。
Ground Truth: Equivalent baseline: Unequivalent ours: Unequivalent	
Sentence pair 9	Somatherapy（或Soma）是一种群体疗法，由威廉·赖希在20 世纪70 年代根据精神分析学家弗莱雷的研究创立。
	Somatherapy（或Soma）是20 世纪70 年代由威尔海姆·赖希根据精神分析学家弗莱雷的研究创立的团体治疗。
Ground Truth: Equivalent baseline: Unequivalent ours: Equivalent	
Sentence pair 10	Sculcoates 拥有一座图书馆、邮局、一个名为Beverley Road Baths 的游泳池、一所高中以及两所小学。
	Sculcoates 有一间图书馆、一间邮局、一个叫做Beverley Road Baths 的游泳浴池、一所小学和两所高中。
Ground Truth: Unequivalent baseline: Equivalent ours: Unequivalent	

due to variations in word order or the presence or absence of certain words. Concretely, "Elizabeth and Waller Cowles" is changed to "Elizabeth Waller and Cowles" in sentence pair 4, and "Raouf will survive" is modified to "Abdul Rahman will survive" in sentence pair 5. On the other hand, CoSENT can predict these sentence pairs accurately. In sentence pair 6, SBERT is misled by wording and format, while CoSENT is successful. For the Chinese group, both models correctly predict that sentence pair 7 is inequivalent. Sentence pair 7 describes the different types of discussions, as the first sentence describes "specific discussion, public image debate" and the second one describes "open discussion, specific file discussion". However, both models fail in predicting sentence 8, which conveys the same basic information and differs in wording (long-term vs. full-time, and outstanding vs. famous). CoSENT outperforms the baseline in sentence pairs 9 and 10. It evaluates sentence pair 9 as equivalent because it only alters the phrasing, and evaluates sentence pair 10 as inequivalent because "one high school and two elementary schools" has been transformed into "one elementary school and two high schools". Both judgments correspond to the ground truth, while the baseline fails.

VI. Conclusion

Cosine similarity is a common measure for sentence embeddings, but directly optimizing it can lead to training failure. Many works strive to design and optimize the encoders so that their induced embedding space fits the cosine-similarity metric. Although these works achieve satisfactory results, they can result in confusion since the cosine function is not involved in the training phase. In this article, we present CoSENT, a novel, consistent, and adaptive sentence embedding framework. Various experiments, including unsupervised and supervised STS tasks for English and Chinese datasets, as well as transfer tasks, demonstrate the effectiveness of our work. CoSENT can be adaptively applied to different types of datasets, e.g., NLI datasets, positive-negative datasets, and scoring datasets. Additionally, we observe that CoSENT converges faster, especially with a speed of 2 times on datasets with hard samples. We expect and explore additional concepts for integrating CoSENT with data enhancement to improve performance. Furthermore, since CoSENT diverges from the typical cross-entropy loss used in interaction-based models, we can explore the use of model blending.

Acknowledgement

This work is supported by the National Key R&D Program of China through grant 2021YFB1714800, NSFC through grant 62322202, Beijing Natural Science Foundation through grant 4222030, Shijiazhuang Science and Technology Plan Project through grant 231130459A, Guangdong Basic and Applied Basic Research Foundation through grant

2023B1515120020, and S&T Program of Hebei through grant 20310101D. Philip S. Yu was supported in part by NSF through grant III-2106758.

References

[1] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung *et al.*, "Multilingual universal sentence encoder for semantic retrieval," in *Proceedings of the ACL*, 2020, pp. 87–94.

[2] P. Zhong and C. Miao, "ntuer at SemEval-2019 task 3: Emotion classification with word and sentence representations in RCNN," in *Proc. of the SemEval*, 2019, pp. 282–286.

[3] L. Chen, F. Wang, R. Yang, F. Xie, W. Wang, C. Xu, W. Zhao, and Z. Guan, "Representation learning from noisy user-tagged data for sentiment classification," *JMLC*, vol. 13, p. 3727–3742, 2022.

[4] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins, "Sparse, dense, and attentional representations for text retrieval," *TACL*, vol. 9, pp. 329–345, 2021.

[5] Y. Hao, X. Liu, J. Wu, and P. Lv, "Exploiting sentence embedding for medical question answering," in *AAAI*, 2019, pp. 938–945.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, vol. 30, 2017, pp. 5998–6008.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[8] K. Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings," in *EMNLP-IJCNLP*, 2019, pp. 55–65.

[9] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," in *EMNLP*, 2020, pp. 9119–9130.

[10] J. Su, J. Cao, W. Liu, and Y. Ou, "Whitening sentence representations for better semantics and faster retrieval," *arXiv preprint arXiv:2103.15316*, 2021.

[11] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *EMNLP*, 2017, pp. 670–680.

[12] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *EMNLP-IJCNLP*, 2019, pp. 3982–3992.

[13] Y. Zhang, J. Baldridge, and L. He, "PAWS: Paraphrase Adversaries from Word Scrambling," in *NAACL*, 2019.

[14] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proc. of the SemEval*, 2017, pp. 1–14.

[15] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A SICK cure for the evaluation of compositional distributional semantic models," in *LREC*, 2014, pp. 216–223.

[16] A. G. Co., "The ATEC semantic similarity learning competition dataset," https://github.com/IceFlameWorm/NLP_Datasets, 2018.

[17] J. Chen, Q. Chen, X. Liu, H. Yang, D. Lu, and B. Tang, "The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification," in *EMNLP*, 2018, pp. 4946–4951.

[18] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *EMNLP*, 2015, pp. 632–642.

[19] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *NAACL-HLT*, 2018, pp. 1112–1122.

[20] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[21] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014, pp. 1188–1196.

[22] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *NeurIPS*, vol. 28, 2015.

[23] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," in *NAACL-HLT*, 2018, pp. 528–540.

[24] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *ICLR*, 2017.

[25] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, "Universal sentence encoder for english," in *Proceedings of EMNLP*, 2018, pp. 169–174.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[27] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*, 2018, pp. 2227–2237.

[28] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *NeurIPS*, vol. 6, 1993.

[29] T. Kim, K. M. Yoo, and S.-g. Lee, "Self-guided contrastive learning for bert sentence representations," in *ACL-IJCNLP*, 2021, pp. 2528–2540.

[30] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," in *ACL-IJCNLP*, 2021, pp. 5065–5075.

[31] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *EMNLP*, 2021, pp. 6894–6910.

[32] Y. Zhang, H. Zhu, Y. Wang, N. Xu, X. Li, and B. Zhao, "A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space," in *ACL*, 2022, pp. 4892–4903.

[33] Y.-S. Chuang, R. Dangovski, H. Luo, Y. Zhang, S. Chang, M. Soljacic, S.-W. Li, S. Yih, Y. Kim, and J. Glass, "DiffCSE: Difference-based contrastive learning for sentence embeddings," in *NAACL-HLT*, 2022, pp. 4207–4218.

[34] J. Liu, J. Liu, Q. Wang, J. Wang, W. Wu, Y. Xian, D. Zhao, K. Chen, and R. Yan, "RankCSE: Unsupervised sentence representations learning via learning to rank," in *ACL*, 2023, pp. 13 785–13 802.

[35] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," in *ACL*, 2020, pp. 6197–6208.

[36] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *EMNLP*, 2020, pp. 6769–6781.

[37] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *IJCAI*, 2017, pp. 4144–4150.

[38] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced lstm for natural language inference," in *ACL*, 2017, pp. 1657–1668.

[39] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in *AAAI*, 2019, pp. 6586–6593.

[40] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, vol. 2, 2006, pp. 1735–1742.

[41] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[42] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *SIMBAD*, 2014.

[43] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2019.

[44] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *CVPR*, 2021, pp. 2495–2504.

[45] L. Ciampiconi, A. Elwood, M. Leonardi, A. Mohamed, and A. Rozza, "A survey and taxonomy of loss functions in machine learning," *arXiv preprint arXiv:2301.05579*, 2023.

[46] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CVPR*, pp. 815–823, 2015.

[47] Y. Feng, H. Wang, D. T. Yi, and R. Hu, "Triplet distillation for deep face recognition," *ICIP*, pp. 808–812, 2019.

[48] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for chinese bert," *TASLP*, pp. 3504–3514, 2021.

[49] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, "Semeval-2012 task 6: A pilot on semantic textual similarity," in *Proc. of the SEM*, 2012, pp. 385–393.

[50] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "SEM 2013 shared task: Semantic textual similarity," in *Proc. SEM*, 2013, pp. 32–43.

[51] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "SemEval-2014 task 10: Multilingual semantic textual similarity," in *Proc. of the SemEval*, 2014, pp. 81–91.

[52] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, L. Uria, and J. Wiebe, "SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability," in *Proc. of the SemEval*, 2015, pp. 252–263.

[53] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe, "SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *Proc. of the SemEval*, 2016, pp. 497–511.

[54] X. Liu, Q. Chen, C. Deng, H. Zeng, J. Chen, D. Li, and B. Tang, "LCQMC:a large-scale Chinese question matching corpus," in *COLING*, 2018, pp. 1952–1962.

[55] Y. Yang, Y. Zhang, C. Tar, and J. Baldridge, "PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification," in *EMNLP*, 2019.

[56] J. Amazonhhh, "A large-scale chinese nature language inference and semantic similarity calculation dataset," https://github.com/pluto-junzeng/CNSD, 2019.

[57] N. Reimers, P. Beyer, and I. Gurevych, "Task-oriented intrinsic evaluation of semantic textual similarity," in *COLING*, 2016, pp. 87–96.

[58] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," in *Proceedings of the LREC*, 2018.

[59] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *ACL*, 2005, pp. 115–124.

[60] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD*. Association for Computing Machinery, 2004, p. 168–177.

[61] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of ACL*, 2004, pp. 271–278.

[62] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, vol. 39, pp. 165–210, 2005.

[63] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, 2013, pp. 1631–1642.

[64] X. Li and D. Roth, "Learning question classifiers," in *COLING*, 2002.

[65] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *COLING*, 2004, pp. 350–356.

Xiang Huang is currently a Master's Degree candidate in the School of Cyber Science and Technology at Beihang University. His research interests include representation learning and natural language processing.

Hao Peng is currently a Professor at the School of Cyber Science and Technology at Beihang University. His current research interests include data mining and deep learning. To date, Dr. Peng has published over 150+ research papers in top-tier journals and conferences, including the IEEE TPAMI, TKDE, TC, TASLP, ACM TOIS, NeurIPS, ICML, SIGIR, and Web Conference. He is the Associate Editor of the JMLC.

Dongcheng Zou is currently a Master's Degree candidate in the School of Cyber Science and Technology at Beihang University. His research interests include deep learning and natural language processing.

Zhiwei Liu is currently a research scientist at Salesforce AI Research. He received Ph.D. degree from University of Illinois Chicago. His current research interests include data mining, natural language processing and representation learning. Dr. Liu has published over 30 research papers in top-tier journals and conferences, including IEEE TKDE, ACM TIST, SIGIR, The Web, EMNLP and etc.

Jianxin Li is currently a Professor with the School of Computer Science and Engineering in Beihang University. His current research interests include machine learning, distributed system, trust management and network security.

Kay Liu is currently a Ph.D. candidate in the Department of Computer Science at the University of Illinois Chicago. His research interests include deep learning and social data mining.

Jia Wu received Ph.D. degree in computer science from the University of Technology Sydney, Australia. Dr. Wu is currently the Research Director for the AI-enabled Processes (AIP) Research Centre and an ARC DECRA Fellow in the School of Computing, Macquarie University, Sydney, Australia. His current research interests include data mining and machine learning. He is the Associate Editor of the ACM Transactions on Knowledge Discovery from Data (TKDD) and Neural Networks (NN).

Jianlin Su is currently an Engineer at Zhuiyi Technology. His current research interests include natural language processing and machine learning.

Philip S. Yu is a Distinguished Professor and the Wexler Chair in Information Technology at the Department of Computer Science, University of Illinois Chicago. He is a Fellow of the ACM and IEEE. Dr. Yu has published more than 1,100 referred conference and journal papers cited more than 160,000 times with an H-index of 180. He has applied for more than 300 patents. Dr. Yu was the Editor-in-chiefs of ACM TKDD (2011-2017) and IEEE TKDE (2001-2004).