Improving Contrastive Learning of Sentence Embeddings from AI Feedback

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, Xipeng Qiu[†]
School of Computer Science, Fudan University
chengqy21@m.fudan.edu.cn

Abstract

Contrastive learning has become a popular approach in natural language processing, particularly for the learning of sentence embeddings. However, the discrete nature of natural language makes it difficult to ensure the quality of positive and negative sample pairs generated through data augmentation methods. Although supervised contrastive learning can produce more accurate sample pairs with human feedback labels, it still lacks fine-grained training signals. In this paper, we propose to improve **C**ontrastive **L**earning of sentence embeddings from **AI F**eedback **(CLAIF)**. Our method utilizes AI feedback from large pre-trained language models (LLMs) to construct sample pairs with fine-grained sample similarity scores to improve contrastive learning. Besides, we combine human feedback and AI feedback to provide better supervision signals for supervised contrastive learning of sentence embeddings. Experimental results show that our method achieves state-of-the-art performance on several semantic textual similarity (STS) and transfer learning tasks compared to other unsupervised and supervised contrastive learning methods. [1]

1 Introduction

Learning sentence embeddings with rich semantics is very important for many natural language processing tasks, such as semantic matching and information retrieval. Recently, pre-trained language models (Devlin et al., 2019; Liu et al., 2019; Qiu et al., 2020) provide a convenient way to get sentence embeddings. However, sentence embeddings directly generated by pre-trained language models show poor performance on semantic textual similarity (STS) tasks due to the representation degeneration problem (Gao et al., 2019). Therefore, finding ways to further improve pre-trained models to produce better sentence embeddings becomes an crucial and fundamental challenge in natural language processing.

Given the shortage of labeled data for sentence embedding learning, recent studies mainly focus on unsupervised methods, such as utilizing contrastive learning methods(Yan et al., 2021; Gao et al., 2021; Chuang et al., 2022). Contrastive learning can be classified into two categories (Khosla et al., 2020): supervised contrastive learning and unsupervised contrastive learning, depending on whether additional label information is utilized to construct positive and negative sample pairs. However, the quality of positive and negative sample pairs in unsupervised contrastive learning can be difficult to ensure. Recent studies also show that data augmentation strategies in unsupervised contrastive learning may introduce some bias like length information (Wu et al., 2022) and improper negatives (Zhou et al., 2022a). While supervised contrastive learning methods can produce more accurate sample pairs by utilizing label information, such as using supervised datasets from natural language inference (Gao et al., 2021), it can only provide coarse-grained labels and lack fine-grained supervision signals. We aruge that these limitations of current contrastive learning methods restrict further performance enhancement of sentence embeddings.

With the emergence of large pre-trained language models (LLMs) (Brown et al., 2020; Sun et al., 2021; Ouyang et al., 2022; Zhang et al., 2022), researchers hope powerful LLMs can help human train other AI models (Bai et al., 2022). One way is to use LLMs to generate datasets using for zero-shot learning (Schick and Schütze, 2021; Ye et al., 2022; Meng et al., 2022). These methods all use predefined labels and task descriptions to generate training inputs, instead of utilizing AI feedback as supervision signals. Therefore, these

[1] We will release our code and data at `https://github.com/xiami2019/CLAIF`

[†] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 11122–11138
July 9-14, 2023 ©2023 Association for Computational Linguistics

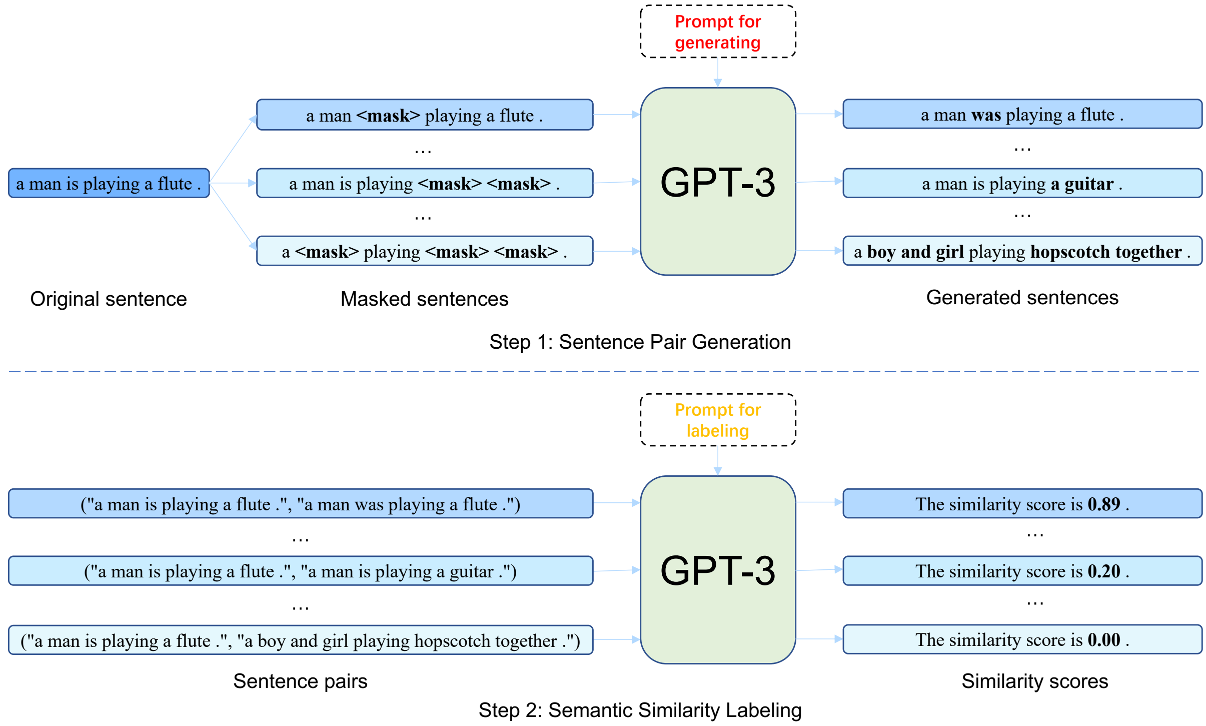
Prompt for generating
a man is playing a flute .
a man <mask> playing a flute .
...
a man is playing <mask> <mask> .
...
a <mask> playing <mask> <mask> .
GPT-3
a man was playing a flute .
...
a man is playing a guitar .
...
a boy and girl playing hopscotch together .
Original sentence
Masked sentences
Generated sentences
Step 1: Sentence Pair Generation
Prompt for labeling
("a man is playing a flute .", "a man was playing a flute .")
...
("a man is playing a flute .", "a man is playing a guitar .")
...
("a man is playing a flute .", "a boy and girl playing hopscotch together .")
GPT-3
The similarity score is 0.89 .
...
The similarity score is 0.20 .
...
The similarity score is 0.00 .
Sentence pairs
Similarity scores
Step 2: Semantic Similarity Labeling

Figure 1: Illustration of the sample pair generation process. The darker the color, the more information the sentence shares with the original sentence.

method are not suitable for tasks whose labels are continuous values and may lead to lack of diversity in training samples. Inspired by these studies, we hope to exploit the capability of LLMs to address shortcomings in contrastive learning of sentence embeddings.

We propose to improve **C**ontrastive **L**earning of sentence embeddings from **AI F**eedback (**CLAIF**). Specifically, we design a two-step sample pair generation method to produce high quality sentence pairs and fine-grained semantic similarity scores using AI feedback from GPT-3, as shown in Figure 1. In the first step, we mask some words in a sentence with different mask rates and then use GPT-3 to generate new sentences based on the remaining information in the masked sentence. Then we combine the generated sentences and the original sentence to construct sentence pairs. In this way, we can use the mask rate to control the amount of sharing information between two sentences in a pair, which will produce sentence pairs with different semantic similarities. In the second step, we utilize GPT-3 to generate semantic similarity scores for sentence pairs. **These scores are the AI feedback on sample similarity.** Since the semantic change caused by reconstructing a masked sentence is difficult to measure, we leverage the linguistic knowledge of LLMs to generate the semantic similarity score. The diversity of AI feedback similarity scores ensured by the sentence pair generation process in the first step. At last we use our generated sample pairs and similarity scores to train the model for sentence embeddings.

In addition to using AI feedback alone, we also combine human feedback and AI feedback by introducing AI feedback into supervised contrastive learning of sentence embeddings which needs human feedback labels to generate positive sample pairs. We use the AI feedback similarity score for the positive sample pair as a soft label to replace the one-hot label in InfoNCE loss (He et al., 2020). We term our loss Soft InfoNCE. This process can be referred to as contrastive learning of sentence embeddings from human and AI feedback (CLHAIF).

We conduct extensive experiments to show the effectiveness of our method. Sentence embeddings learned with CLAIF and CLHAIF achieve state-of-the-art performance on standard semantic textual similarity tasks and outperform strong baselines on transfer learning tasks. We also find that CLAIF results in significant improvements to the cross-encoder architecture for the sentence-pair modeling task.

Our main contributions are as follows:

Feedback Source	Positive Pair	Negative Pair	Loss Function
Zero Feedback (CLZF)	(x_i, x_i')	$\{(x_i, x_j) \mid x_j \in X, i \neq j\}$	InfoNCE (van den Oord et al., 2018; He et al., 2020; Gao et al., 2021), NT-Xent (Chen et al., 2020)
Human Feedback (CLHF)	(x_i, x_i^+)	$\{(x_i, x_i^-), (x_i, x_j) \mid x_j \in X, i \neq j\}$	SupCon (Khosla et al., 2020), InfoNCE (Gao et al., 2021), KNN-Contrastive (Zhou et al., 2022b)
AI Feedback (CLAIF)	(x_i, x_i', y_i)	$(x_i, x_i', y_i)^*$	Mean Squared Error
Human and AI Feedback (CLHAIF)	(x_i, x_i^+, y_i)	$\{(x_i, x_i^-), (x_i, x_j) \mid x_j \in X, i \neq j\}$	Soft InfoNCE

Table 1: The details of contrastive learning from different feedback. X is the full set containing all samples and x_i is the i-th sample of X, such as a sentence or an image. x_i' is an augmented sample obtained by using some data augmentation strategies to x_i. x_i^+ and x_i^- are postive sample and negative sample of x_i picked by human feedback information, such as class label information. y_i is the AI feedback sample similarity score for the i-th sample pair. $*$: CLAIF does not explicitly construct positive and negative pairs, sample pairs with high simiarity scores can be seen as positive pairs and those with low scores can be seen as negative pairs.

- We propose to improve contrastive learning of sentence embeddings from AI feedback (CLAIF) and achieve state-of-the-art performance on several semantic textual similarity tasks and transfer learning tasks.

- We construct a semantic textual similarity dataset with high quality sentence pairs and fine-grained AI feedback similarity scores using large pre-trained language models.

- We propose a method to incorporate human feedback and AI feedback to provide better supervision for contrastive learning of sentence embeddings.

- Experimental results show the scalability of CLAIF, which is cheaper and more efficient than collecting data from human feedback.

2 Understanding Contrastive Learning from Different Feedback

In this section, we categorize contrastive learning methods into four categories according to their feedback sources. We summarize the details of contrastive learning from different feedback in Table 1, including their feedback types, sample pairs construction methods and representative loss functions.

2.1 Contrastive Learning from Zero Feedback

Traditional contrastive learning is used for self-supervised representation learning (Hadsell et al., 2006; He et al., 2020). These methods construct positive and negative sample pairs using data augmentation strategies without any human feedback. For example, in natural language processing, Gao et al. (2021) construct positive sample pairs by doing the dropout operation twice for the same sentence and negative pairs by combining with another sentences. We refer to these methods as Contrastive Learning from Zero Feedback (CLZF). The most common loss function for CLZF is InfoNCE (van den Oord et al., 2018). Chen et al. (2020) propose NT-Xent loss, which can be seen as a variant of InfoNCE. However, due to the discrete nature of natural language, it is hard to find effective and unbiased data augmentation strategies to construct high quality sample pairs.

2.2 Contrastive Learning from Human Feedback

Recently, Khosla et al. (2020) propose to use label information to construct positive sample pairs. In sentence embeddings, Gao et al. (2021) use premise-hypothesis pairs with entailment relationship from natural language inference (NLI) datasets as positive sample pairs and still use InfoNCE for training. Since these methods leverage label information from human, we refer to them as Contrastive Learning from Human Feedback (CLHF). With the help of label information, some new losses can be used in CLHF, like SupCon (Khosla et al., 2020) and KNN-Contrastive (Zhou et al., 2022b). Although CLHF can construct more accurate sample pairs, it still lacks fine-grained supervision signals. For example, in InfoNCE, all positive pairs have a label of 1. But there are also differences in the similarity between different positive sample pairs.

2.3 Contrastive Learning from AI Feedback

Measuring the similarity of sample pairs in contrastive learning is a laborious task. However, thanks to emergence of LLMs, we can use LLMs to measure the similarity of sample pairs and use the AI feedback as our training signals. We refer to this approach as Contrastive Learning from AI Feedback (CLAIF). CLAIF does not need to explicitly construct positive and negative sample pairs because each sample pair has a fine-grained label. We use mean squared error (MSE) loss for the training of CLAIF in this work.

2.4 Contrastive Learning from Human and AI Feedback

Besides contrastive learning from AI feedback, we propose to combine human and AI feedback to produce better supervision signals when they are both available. We call this category contrastive learning from human and AI feedback (CLHAIF) and we propose a soft InfoNCE loss for the training of CLHAIF. We hope to use fine-grained AI feedback to refine the coarse-grained signals in current CLHF methods.

3 Methodology

In this section, we first introduce our method to generate sample pairs and the training process of CLAIF. In order to obtain high quality sentence pairs with diverse and fine-grained similarity scores, we propose a two-step sample pair generation method: **Sentence Pair Generation** and **Semantic Similarity Labeling**. The generation process is shown in Figure 1. We use these sample pairs to train language models like BERT and RoBERTa. Then we introduce CLHAIF, which combines human and AI feedback in contrastive learning of sentences embeddings.

3.1 Sentence Pair Generation

We use unpaired sentences from the training set of STS Benchmark (Cer et al., 2017) as our original sentences. As shown in Figure 1, we first mask some words of the original sentence *"a man is playing a flute."* with different mask rates using the `<mask>` token, in order to delete some information in the original sentence. The more words that are masked, the less information is left. We use the depth of color to indicate the degree of information sharing between two sentences in Figure 1. Then we write a task description prompt to steer GPT-3 to generate new sentences based on masked sentences. We provide our task descriptions in Appendix B. To increase the diversity of generated sentences, we merge adjacent `<mask>` tokens in 50% of masked sentences into one `<mask>` token. Then we combine the original sentence with each generated sentence to construct sentence pairs.

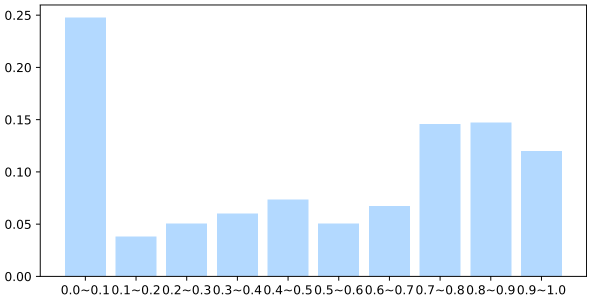

Figure 2: The score distribution of our generated sample pairs. The x-axis is the similarity score and the y-axis is the percentage of the score.

3.2 Semantic Similarity Labeling

In this step, we label the semantic similarity score for each sentence pair using AI feedback from GPT-3. The similarity score ranges from 0 to 1, where a score of 1 means that the semantic of the two sentences are exactly the same, and a score of 0 means that the semantic of the two sentences are completely different. We write a task description prompt to steer GPT-3 to generate a similarity score between 0 and 1 for each sample pair generated in step 1. The first step ensures the diversity of semantic similarity scores. As illustrated in Figure 2, the generated scores are diverse and distributed in the value range from 0 to 1.

3.3 Training on Generated Pairs

With the generated sample pairs, we train a language model as the sentence encoder to get better sentence embeddings. Given diverse sentence pairs which have fine-grained similarity scores, we do not need to explicitly construct positive and negative sample pairs. Therefore, we directly use the mean squared error (MSE) loss to fit the cosine similarity of each sentence pair to its AI feedback similarity score:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left[\cos\left(\mathbf{h}_i, \mathbf{h}_i'\right) - y_i\right]^2 \tag{1}$$

where N is the batch size, $\mathbf{h}_i$ and $\mathbf{h}_i'$ are two sentence embeddings of the i-th sentence pair (x_i, x_i') encoded by the model, y_i is the corresponding similarity score and cos means the calculation of cosine similarity. During inference, we use the cosine similarity of two sentence embeddings as their semantic similarity score.

3.4 Combining Human Feedback and AI Feedback

In this section, we mainly study the cooperation of human and AI models to provide better training signals for contrastive learning, which we called CLHAIF. Reimers and Gurevych (2019) use supervised NLI datasets to learn sentence embeddings. Gao et al. (2021) construct positive and hard negative sample pairs for contrastive learning leveraging label information of NLI datasets, achieving significant improvements. However, as we mentioned in Section 2.2, CLHF does not distinguish between different positive sample pairs and assigns label of 1 for all positive pairs. In this way, all positive sample pairs are pulled together with the same extent in contrastive learning, ignoring differences in similarity between different positive pairs. Therefore, we use AI feedback to refine these coarse-grained supervision signals.

At first, we use the semantic similarity labeling step in Section 3.2 to generate AI feedback similarity scores for sentence pairs constructed from supervised NLI datasets: SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018). Following Gao et al. (2021), we construct sample pairs using the label information. For the i-th sample of the NLI dataset, we can obtain two sentence pairs (x_i, x_i^+) and (x_i, x_i^-), where x_i is the premise, x_i^+ and x_i^- are entailment and contradiction hypothesis. (x_i, x_i^+) is the positive pair and (x_i, x_i^-) is the hard negative pair.

In order to incorporate AI feedback, we propose soft InfoNCE loss by replacing the one-hot label with the AI feedback score as the soft label:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} l_i \tag{2}$$

$$l_i = y_i \log \frac{e^{\cos(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\cos(\mathbf{h}_i, \mathbf{h}_j^+)/\tau} + e^{\cos(\mathbf{h}_i, \mathbf{h}_j^-)/\tau}\right)}$$

where N is the batch size, $\mathbf{h}_i$, $\mathbf{h}_i^+$ and $\mathbf{h}_i^-$ are sentence embeddings of x_i, x_i^+ and x_i^-, y_i is the AI feedback similarity score for the positive pair (x_i, x_i^+) and τ is the temperature parameter.

4 Experiments

4.1 Evaluation Datasets

We conduct extensive experiments on seven semantic textual similarity (STS) tasks and seven transfer learning tasks. The STS tasks include STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). The transfer learning tasks include MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000) and MRPC (Dolan and Brockett, 2005).

Following Gao et al. (2021), for STS tasks, we calculate the Spearman's correlation between the cosine similarity of sentence embeddings and the golden similarity scores from STS datasets. For transfer learning tasks, we train a logistic regression classifier based on fixed sentence embeddings and follow the default settings of SentEval (Conneau and Kiela, 2018). We use the same evaluation script as Gao et al. (2021) to calculate metrics.

4.2 Baselines

We compare our method with some strong baselines among three types of sentence embedding methods:

Post-processing methods: These methods adopt some post-processing operations to enhance sentence embeddings which do not need to further train the backbone model. We use BERT-whitening (Su et al., 2021), BERT-flow (Li et al., 2020) and prompt based BERT (Jiang et al., 2022) as baselines.

Training methods: These methods use additional data to further train the backbone model for better sentence embeddings. We use SBERT (Reimers and Gurevych, 2019), ConSERT (Yan et al., 2021),

Dataset	Sample Number	Sample Type
Wiki-1M	1,000,000	sentence
NLI	275,601	sentence triplet
Dino	83,497	sentence pair
CLAIF	113,773	sentence pair
CLAIF$_{\text{scaled}}$	1,215,618	sentence pair

Table 2: Statistics of datasets for different settigns. Wiki-1M is used by CLZF methods. NLI is used by CLHF methods. We use CLAIF and CLAIF$_{\text{scaled}}$ to refer to our generated datasets here.

SimCSE (Gao et al., 2021), DiffCSE (Chuang et al., 2022) and PromptBERT (Jiang et al., 2022) as baselines.

Dataset-generation based methods: Some studies generate datasets from LLMs for sentence embedding learning. We use Dino (Schick and Schütze, 2021) as our baseline. Dino generates sentence pairs based on three discrete similarity labels using GPT2-XL. For a fair comparison, we re-implement Dino using GPT-3 in our experiments.

4.3 Implementation Details

Choice of large pre-trained language models: In our experiments, we get all AI feedback from text-davinci-003, which is the latest version of GPT-3. We access text-davinci-003 through the OpenAI API.

Sample pair generation: We use nine mask rates for each original sentence in sentence pair generation: *0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8*. For CLAIF, we use unpaired sentences from the training set of STS-B as original sentences to construct sentence pairs from scratch and randomly sample two other sentences for each original sentence to construct two sentence pairs with a similarity score of 0. For CLHAIF, following previous studies (Gao et al., 2021; Jiang et al., 2022), we use the SNLI and MNLI datasets to construct sentence pairs and add a AI feedback similarity score for each sentence pair. We only use the AI feedback scores for positive pairs in our experiments of CLHAIF. Besides, to demonstrate the scalability of CLAIF, we use sentence pairs constructed from STS-B and from NLI datasets for the training of CLAIF, which we called CLAIF$_{\text{scaled}}$. We list statistics of some datasets used for different methods in Table 2.

Training details: We use the base version of the pre-trained language model BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as our backbone models. We use the development set of STS-B as our validation set. In CLAIF, we use the mean pooling strategy to get sentence embeddings for BERT and RoBERTa. For CLHAIF, we take the same pooling strategy as the corresponding baseline. Other implementation details are in Appendix A.

Model	SentEval Avg.
SimCSE$_{\text{BERT}}$	85.81
PromptBERT	85.49
DiffCSE$_{\text{BERT}}$	**86.86**
CLAIF$_{\text{BERT}}$	86.62
SimCSE$_{\text{RoBERTa}}$	84.84
PromptRoBERTa	87.36
DiffCSE$_{\text{RoBERTa}}$	87.04
CLAIF$_{\text{RoBERTa}}$	**87.99**
SimCSE$_{\text{BERT-supervised}}$	86.51
w/ CLHAIF	86.73
PromptBERT$_{\text{supervised}}$	86.98
w/ CLHAIF	**87.09**
SimCSE$_{\text{RoBERTa-supervised}}$	88.08
w/ CLHAIF	88.82
PromptRoBERTa$_{\text{supervised}}$	89.11
w/ CLHAIF	**89.27**
CLAIF$_{\text{scaled-BERT}}$	87.15
CLAIF$_{\text{scaled-RoBERTa}}$	**89.44**

Table 3: The performance comparison of CLAIF and CLHAIF on transfer learning tasks. SentEval Avg is the average accuracy on seven transfer learning datasets from SentEval.

4.4 Main Results

Semantic Textual Similarity We compare CLAIF with methods which do not use additional labeled datasets for training, including CLZF methods and dataset generation methods. The results of CLAIF on STS tasks are shown in Table 4. We observe that CLAIF achieves the best performance on the four datasets STS15, STS16, STS-B, SICK-R and get the highest averaged Spearman's correlation on seven STS datasets. And in the comparison with dataset generation methods, CLAIF outperforms Dino by 3.37 and 2.75 points on BERT and RoBERTa. Therefore, we believe that CLAIF is more effective for the learning of sentence embeddings than CLZF methods.

We implement CLHAIF by incorporating AI feedback into supervised SimCSE and supervised PromptBERT/PromptRoBERTa. We compare CLHAIF with other methods that use additional labeled datasets for training. As shown in Table 5, incorporating AI feedback improves results of

Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg.**
				BERT-base				
BERT-flow[†]	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT-whitening[†]	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
Prompt based BERT[†]	60.96	73.83	62.18	71.54	68.68	70.60	67.16	67.85
ConSERT[†]	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE[†]	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
DiffCSE[‡]	72.28	84.43	76.47	83.90	80.54	80.59	71.23	78.49
PromptBERT[†]	71.56	**84.58**	**76.98**	84.47	80.60	81.60	69.87	78.54
Dino$_{\text{GPT-3}}$	**72.61**	81.92	75.09	80.42	76.26	77.10	70.43	76.26
CLAIF	70.62	81.51	76.29	**85.05**	**81.36**	**84.34**	**78.22**	**79.63**
				RoBERTa-base				
RoBERTa-whitening[†]	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
SimCSE[†]	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
DiffCSE[‡]	70.05	83.43	75.49	82.81	82.12	82.38	71.19	78.21
PromptRoBERTa[†]	**73.94**	**84.74**	**77.28**	84.99	81.74	81.88	69.50	79.15
Dino[§]	70.27	81.26	71.25	80.49	77.18	77.82	68.09	75.20
Dino$_{\text{GPT-3}}$	71.24	81.55	75.67	81.42	78.77	80.10	71.31	77.15
CLAIF	68.33	82.26	77.00	**85.18**	**83.43**	**85.05**	**78.02**	**79.90**

Table 4: The performance comparison of CLAIF on STS tasks. †: results from (Jiang et al., 2022). ‡: results from (Chuang et al., 2022). §: results from (Schick and Schütze, 2021). Other results are from our experiments. We bold the highest results among models with the same backbone.

Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg.**
				BERT-base				
SBERT[†]	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT-flow[†]	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
SBERT-whitening[†]	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
ConSERT[†]	74.07	83.93	77.05	83.66	78.76	81.36	76.77	79.37
SimCSE[†]	**75.30**	84.67	80.19	85.40	80.82	84.25	80.39	81.57
w/ CLHAIF	$74.86_{\downarrow 0.44}$	$85.09_{\uparrow 0.42}$	$81.24_{\uparrow 1.05}$	$85.96_{\uparrow 0.56}$	$81.33_{\uparrow 0.51}$	$84.96_{\uparrow 0.71}$	$\mathbf{81.36}_{\uparrow 0.97}$	$82.08_{\uparrow 0.51}$
PromptBERT*	75.10	85.54	80.58	86.00	81.24	84.57	80.36	81.91
w/ CLHAIF	$75.03_{\downarrow 0.07}$	$\mathbf{85.88}_{\uparrow 0.34}$	$\mathbf{81.48}_{\uparrow 0.90}$	$86.33_{\uparrow 0.33}$	$81.40_{\uparrow 0.16}$	$84.93_{\uparrow 0.36}$	$80.98_{\uparrow 0.62}$	$82.29_{\uparrow 0.38}$
CLAIF$_{\text{scaled}}$	74.36	85.07	80.64	**87.21**	**83.36**	**86.26**	79.68	**82.37**
				RoBERTa-base				
SRoBERTa[†]	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa-whitening[†]	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
SimCSE[†]	**76.53**	85.21	80.95	86.03	82.57	85.83	80.50	82.52
w/ CLHAIF	$76.23_{\downarrow 0.30}$	$85.46_{\uparrow 0.25}$	$81.48_{\uparrow 0.53}$	$86.47_{\uparrow 0.44}$	$83.40_{\uparrow 0.83}$	$85.93_{\uparrow 0.10}$	$\mathbf{80.95}_{\uparrow 0.45}$	$82.85_{\uparrow 0.33}$
PromptRoBERTa*	76.41	85.64	82.11	86.18	82.71	85.74	79.95	82.68
w/ CLHAIF	$76.26_{\downarrow 0.15}$	$\mathbf{86.01}_{\uparrow 0.37}$	$\mathbf{82.83}_{\uparrow 0.72}$	$86.70_{\uparrow 0.52}$	$82.94_{\uparrow 0.23}$	$\mathbf{86.04}_{\uparrow 0.30}$	$80.55_{\uparrow 0.60}$	$\mathbf{83.05}_{\uparrow 0.37}$
CLAIF$_{\text{scaled}}$	72.58	84.50	79.48	**86.92**	**84.19**	85.85	79.64	81.88

Table 5: The performance comparison of CLHAIF on STS tasks. †: results from Jiang et al. (2022). Other results are from our experiments. ∗: The results of PromptBERT and PromptRoBERTa are obtained by running official code of Jiang et al. (2022) with recommended hyperparameters.

CLHF methods like supervised SimCSE on six STS datasets except STS12.

Transfer Tasks In addition to STS tasks, we also evaluate several transfer learning tasks from SentEval. Experimental results show that sentence embeddings learned with CLAIF and CLHAIF also achieve better or comparable performance compared to baselines. We present the average results for seven transfer tasks in Table 3 and detailed results in Appendix C.

4.5 Scalability of CLAIF

In this section we discuss the scalability of CLAIF. The results of CLAIF$_{\text{scaled}}$ in Table 5 show that using more data to scale CLAIF can bring significant improvements. CLAIF$_{\text{scaled}}$ greatly outputforms CLAIF by 2.74 points on BERT-base (79.63 → 82.37) and even outputforms or performs on par with CLHF and CLHAIF methods. We believe that using more data can further improve the performance of CLAIF. Since collecting data from AI

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
				BERT-base				
Trans-Encoder$_{\text{cross}}$	71.94	84.14	76.39	82.87	80.65	81.06	71.16	78.32
CLAIF$_{\text{cross}}$	70.36	83.27	79.73	87.87	84.54	85.00	78.33	81.30
				RoBERTa-base				
Trans-Encoder$_{\text{cross}}$	72.59	83.24	76.83	84.20	82.82	82.85	69.51	78.86
CLAIF$_{\text{cross}}$	72.80	83.75	81.52	88.66	86.61	87.05	81.28	83.10

Table 6: The performance comparison of CLAIF based on the cross-encoder architecture.

feedback is more cheaper than from human feedback, we argue that CLAIF has great potential in practical applications.

4.6 Sentence-Pair Modeling

In this section, we evaluate CLAIF on the sentence-pair modeling task. Cross-encoders usually outperform bi-encoders in information retrieval. However, we observe in Liu et al. (2022) that the cross-encoder does not show its superior on sentence-pair modeling. We contribute this to the lack of fine-grained training signals. We train a cross-encoder with CLAIF. Experimental results in Table 11 show that, with the help of AI feedback, CLAIF$_{\text{cross}}$ brings significant improvements for cross-encoders on the sentence-pair modeling task compared to the previous model Trans-Encoder (Liu et al., 2022). More training details are in Appendix D.

4.7 Human Evaluation

In this section, we conduct human evaluation to measure the quality of generated sentences and similarity scores. We measure whether the generated sentences are fluent and whether the similarity scores are consistent with the real semantic similarities. To help human judge the consistency, we generate a natural language explanation for each generated similarity score using GPT-3. We invite 4 experts to participate in our human evaluation. Then we random pick 100 samples from the dataset used in CLAIF and assign 25 samples to each expert. In the evaluation, 92 percent of generated sentences are considered fluent and 90 percent of generated scores are considered consistent by the expert, which means our method can generate high quality sentence pairs and similarity scores.

5 Related Work

Recent studies about sentence embeddings mainly focuse on using additional data to further train pre-trained language models. Yan et al. (2021) and Gao et al. (2021) propose different data augmentation strategies for contrastive learning and achieve significant improvements using unlabeled data. Chuang et al. (2022) use equivariant contrastive learning for learning better representations. Zhou et al. (2022a) and Wu et al. (2022) address the bias caused by construction processes of negative and positive samples. Jiang et al. (2022) use different prompt templates to produce positive pairs for contrastive learning. Opitz and Frank (2022) use various semantic sentence features to construct fine-grained labels for sentence embedding training.

Impressed by the powerful capabilities of LLMs (Brown et al., 2020; Ouyang et al., 2022), researchers pay more attention to using AI feedback from LLMs for zero-shot and few-shot learning. Li et al. (2023); Li and Qiu (2023) use AI feedback from language models to enhance In-context Learning and Chain-of-Thoughts. Ye et al. (2022) and Meng et al. (2022) generate datasets by taking labels and prompts as the input of LLMs and then let LLMs generate training samples. Schick and Schütze (2021) design a dataset generation method for STS tasks. They construct three natural language instructions based on three discrete similarity scores and then use these instructions to steer LLMs to construct sentence pairs. However, it is hard to use natural language to describe various similarity scores, since the similarity score is a continuous variable with values ranging from 0 to 1.

6 Conclusion

In this paper, we first formalize four types of contrastive learning: contrastive learning from zero feedback (CLZF), contrastive learning from human feedback (CLHF), contrastive learning from AI feedback (CLAIF) and contrastive learning from human and AI feedback (CLHAIF). Then we improve contrastive learning of sentence embeddings from AI feedback and combine human feedback with AI feedback to produce better supervision signals. Experimental results show that CLAIF and CLHAIF can bring substantial improvements for sentence embedding learning. We hope that learning from AI feedback can shed new lights for representation learning and contrastive learning.

Limitations

To inspire future work, we conclude some limitations of our work as follows:

- While our method achieves promising performance on sentence embedding related tasks like STS, the performance on other natural language processing tasks are still need to investigate.
- The AI feedback in our experiments comes from GPT-3, which requires a fee to use.
- We do not explore the effect of different task description prompts on the quality of generated sample pairs, which may influence the performance of CLAIF.
- In CLHAIF, we only use the AI feedback for positive sample pairs. How to utilize AI feedback for negative sample pairs remains to be studied.

Acknowledgement

We would like to thank all anonymous reviewers for their valuable advice. This work was supported by the National Natural Science Foundation of China (No. 62236004 and No. 62022027).

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. 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@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 252–263. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. 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@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 81–91. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. 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@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 497–511. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 385–393. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 32–43. Association for Computational Linguistics.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosiute, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemí Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional AI: harmlessness from AI feedback. *CoRR*, abs/2212.08073.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The 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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James R. Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4207–4218. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*. Asian Federation of Natural Language Processing.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.

Ting Jiang, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. 2022. Promptbert: Improving BERT sentence embeddings with prompts. *CoRR*, abs/2201.04337.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9119–9130. Association for Computational Linguistics.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning.

Xiaonan Li and Xipeng Qiu. 2023. Mot: Pre-thinking and recalling enable chatgpt to self-improve with memory-of-thoughts.

Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. 2022. Trans-encoder: Unsupervised sentence-pair modelling through self- and mutual-distillations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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. *CoRR*, abs/1907.11692.

Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, and Haifeng Wang. 2022. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *CoRR*, abs/2205.09153.

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 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *CoRR*, abs/2202.04538.

Juri Opitz and Anette Frank. 2022. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 625–638. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, pages 271–278. ACL.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 115–124. The Association for Computer Linguistics.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6943–6951. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316.

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. *CoRR*, abs/2107.02137.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 296–310. Association for Computational Linguistics.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 24-28, 2000, Athens, Greece*, pages 200–207. ACM.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Lang. Resour. Evaluation*, 39(2-3):165–210.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

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, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 3898–3907. 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. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5065–5075. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *CoRR*, abs/2202.07922.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Kun Zhou, Beichen Zhang, Xin Zhao, and Ji-Rong Wen. 2022a. Debiased contrastive learning of unsupervised sentence representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6120–6130. Association for Computational Linguistics.

Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 2022b. KNN-contrastive learning for out-of-domain intent classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5129–5141, Dublin, Ireland. Association for Computational Linguistics.

A Implementation Details

For CLAIF, we train our models for 3 epochs with a batch size of 32, and set the learning rate to 2e-5. Following previous work, we use the development set of STS-B as the validation set. We evaluate the model every 125 training steps on the validation set to choose the best checkpoint during training. We conduct a grid-search of learning rate $\in \{$1e-5,2e-5$\}$ on the validation set.

For CLHAIF, we use the official implementation and the default configuration of our baselines SimCSE (Gao et al., 2021) and PrompBERT (Jiang et al., 2022). We only replace the one-hot label with our soft label.

We run experiments of CLAIF on a single RTX 3090 GPU with 24G gpu memory and experiments of CLHAIF on 4 RTX 3090 GPUs. We fix the random seed to 42 for all experiments.

B Task Descriptions

We use three task description prompts in our experiments. For sentence pair generation in Section 3.1, our two prompts are:

"Replace all <mask> tokens in '<masked-sentence>' to make a new sentence. The new sentence is:" and *"Write two sentences that mean the same thing. Sentence 1: '<sentence1>' Sentence 2:"*.

For semantic similarity labeling in Section 3.2, our prompt is:

"The similarity score for two sentences is in the range from 0.0 to 1.0, 0.0 means completely different and 1.0 means almost the same. Now given two sentences '<sentence1>' and '<sentence2>', please give a similarity score for these two sentences: The similarity score for these two sentences is".

C Transfer Learning Tasks

We list the detailed performance comparison of CLAIF and CLHAIF in Table 7 and Table 8. Experimental results show that CLAIF achieves the best performance on RoBERTa-base and comparable performance on BERT-base. CLHAIF also achieves better results compared to the baselines. Using more data to scale CLAIF also brings performance improvements on transfer learning tasks as shown in Tabel 8.

Model	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
				BERT-base				
Avg. BERT embeddings†	78.66	86.25	94.37	88.66	84.40	**92.80**	69.54	84.94
BERT- [CLS] embedding†	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
SimCSE‡	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
SimCSE w/MLM‡	**82.92**	87.23	**95.71**	88.73	**86.81**	87.01	**78.07**	86.64
DiffCSE‡	82.69	87.23	95.23	89.28	86.60	90.40	76.58	**86.86**
PromptBERT†	80.74	85.49	93.65	89.32	84.95	88.20	76.06	85.49
Dino$_{\text{GPT-3}}$	79.96	85.27	93.67	88.87	84.29	88.60	69.62	84.33
CLAIF	81.64	**87.98**	94.24	**89.34**	86.16	89.80	77.16	86.62
				RoBERTa-base				
Avg. RoBERTa embeddings	**84.35**	88.34	**95.28**	86.13	89.46	**93.20**	74.20	87.28
SimCSE‡	81.04	87.74	93.28	86.94	86.60	84.60	73.68	84.84
SimCSE w/MLM‡	83.37	87.76	95.05	87.16	89.02	90.80	75.13	86.90
DiffCSE‡	82.82	88.61	94.32	87.71	88.63	90.40	76.81	87.04
PromptRoBERTa†	83.82	88.72	93.19	**90.36**	88.08	90.60	76.75	87.36
Dino$_{\text{GPT-3}}$	82.31	88.66	93.95	88.72	87.53	88.20	73.74	86.16
CLAIF	84.11	**90.62**	94.29	89.13	**89.57**	91.00	**77.22**	**87.99**

Table 7: The performance comparison of CLAIF on transfer learning tasks. †: results from (Jiang et al., 2022). ‡: results from (Chuang et al., 2022). Other results are from our experiments.

D Sentence-Pair Modeling

In sentence-pair modeling task, cross-encoders can be used to directly encode the sequence of two sentences and then predict a similarity score. Previous studies (Thakur et al., 2021; Liu et al., 2022; Lu et al., 2022) show that cross-encoders usually outperform bi-encoders. We find that CLAIF can significantly improve the performance of cross-encoders on sentence-pair modeling task, with the help of fine-grained AI feedback scores.

We use the binary cross-entropy (BCE) loss to train cross-encoders initialized from BERT and RoBERTa:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} l_i \tag{3}$$

$$l_i = y_i \log \sigma(\hat{y}_i) + (1 - y_i) \log (1 - \sigma(\hat{y}_i))$$

where N is the batch size, $\hat{y}_i$ is the predicted score of the i-th sentence pair, y_i is the AI feedback similarity score and σ is the sigmoid function.

E Cost for Data Generation

According to our billings, we spent about $100 to generate data for CLAIF and about $720 for the scaled dataset.

F Generated Examples

We present some generated sample pairs used in CLAIF in Table 9 and some generated similarity scores for sample pairs constructed from NLI in Table 10.

G Comparison with Text-Ada-Embedding-002

Recently, OpenAI has released a powerful embedding model named text-ada-embedding-002, we compare the performance of it on STS tasks with CLAIF here. The results show that CLAIF-scaled achieves better performance on STS tasks than text-ada-embedding-002.

Model	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
				BERT-base				
SBERT†	**83.64**	**89.43**	94.39	89.86	**88.96**	89.60	76.00	**87.41**
SimCSE†	82.69	89.25	**94.81**	89.59	87.31	88.40	73.51	86.51
w/ CLHAIF	$83.11_{\uparrow 0.42}$	$88.98_{\downarrow 0.27}$	$94.47_{\downarrow 0.34}$	$89.95_{\uparrow 0.36}$	$88.58_{\uparrow 1.27}$	$86.40_{\downarrow 2.00}$	$75.65_{\uparrow 2.14}$	$86.73_{\uparrow 0.22}$
PromptBERT*	83.05	88.96	94.68	89.86	88.19	87.80	76.29	86.98
w/ CLHAIF	$83.14_{\uparrow 0.09}$	$89.12_{\uparrow 0.16}$	$94.65_{\downarrow 0.03}$	$89.97_{\uparrow 0.11}$	$87.86_{\downarrow 0.33}$	$88.80_{\uparrow 1.00}$	$76.06_{\downarrow 0.23}$	$87.09_{\uparrow 0.11}$
CLAIF$_{\text{scaled}}$	82.08	89.12	94.48	**90.22**	87.53	**90.20**	**76.41**	87.15
				RoBERTa-base				
SRoBERTa†	84.91	90.83	92.56	88.75	90.50	88.60	78.14	87.76
SimCSE†	84.92	**92.00**	94.11	89.82	91.27	88.80	75.65	88.08
w/ CLHAIF	$86.10_{\uparrow 1.18}$	$91.76_{\downarrow 0.24}$	$94.66_{\uparrow 0.55}$	$90.07_{\uparrow 0.25}$	$91.93_{\uparrow 0.66}$	$91.60_{\uparrow 2.80}$	$75.59_{\downarrow 0.06}$	$88.82_{\uparrow 0.74}$
PromptRoBERTa*	86.22	91.55	**95.08**	90.97	91.82	91.40	76.70	89.11
w/ CLHAIF	**86.41**$_{\uparrow 0.19}$	$91.76_{\uparrow 0.21}$	$94.90_{\downarrow 0.18}$	**91.01**$_{\uparrow 0.04}$	**92.04**$_{\uparrow 0.22}$	$92.40_{\uparrow 1.00}$	$76.35_{\downarrow 0.35}$	$89.27_{\uparrow 0.16}$
CLAIF$_{\text{scaled}}$	85.05	91.71	94.39	90.03	91.87	**94.00**	**79.01**	**89.44**

Table 8: The performance comparison of CLHAIF on transfer learning tasks. †: results from Jiang et al. (2022). $*$: The results of PromptBERT and PromptRoBERTa are obtained by running official code of Jiang et al. (2022) with recommended hyperparameters.

Original Sentence	Generated Sentence	Similarity Score
a plane is taking off .	an aircraft is departing .	0.80
	The airplane is taking off.	0.80
	A plane is taking off swiftly	0.90
	The blue plane is taking off.	0.75
	Airplane is flying.	0.67
	Bob and Joe are taking a walk.	0.00
	Aeroplane is flying	0.67
	Put off steam	0.00
	Turn off lights	0.00
a man is playing a large flute .	A male individual is performing on a big flute.	0.86
	a man is playing a large flute.	1.00
	He she is playing a large flute.	0.78
	a man played a wooden flute.	0.71
	a flute is not a wooden flute	0.20
	a boy playing a large drum	0.33
	a man is wise.	0.00
	The old man stood .	0.00
	The quick brown fox jumps over the lazy dog	0.00
three men are playing chess .	There are three men playing chess.	0.94
	Three children are playing chess.	0.80
	Three kings are playing chess.	0.87
	They are playing chess .	0.80
	three men played chess together	0.78
	three men are walking	0.00
	John and Mary were playing chess together	0.50
	I play blitz chess online	0.20
	I like to play soccer and tennis.	0.00

Table 9: Generated examples of sample pairs used in CLAIF.

Premise	**Entailment Hypothesis**	**Similarity Score**
The other men shuffled.	The other men were shuffled around.	0.78
well it's been very interesting	It has been very intriguing.	0.90
He started slowly back to the bunkhouse.	He returned slowly to the bunkhouse.	0.91
well what the market can bear and	The market can bear some.	0.71
She smiled back.	She was happy.	0.25
The economy could be still better.	It still have room for improvement.	0.55
The man should have died instantly.	The man should not have been alive.	0.14
Turned out, I wasn't completely wrong.	I was not totally wrong.	0.8

Table 10: Generated examples of similarity scores used in CLHAIF.

Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg.**
Ada-Embedding-002	69.80	83.26	76.08	86.12	**85.96**	84.30	**80.25**	80.82
CLAIF-BERT	70.62	81.51	76.29	85.05	81.36	84.34	78.22	79.63
CLAIF-BERT$_{\text{scaled}}$	**74.36**	**85.07**	**80.64**	**87.21**	83.36	**86.26**	79.68	**82.37**

Table 11: The performance comparison between CLAIF and OpenAI's text-ada-embedding-002.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations.

☒ A2. Did you discuss any potential risks of your work?
Our work is about representation learning and contrastive learning, which are general methods and do not have potential risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, 1 Introduction section.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Implementation Details section in Appendix.

☑ B1. Did you cite the creators of artifacts you used?
Implementation Details section in Appendix.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Because we only use the public datasets and open source code in this work.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Our use of these public datasets and open source code is exactly what it was intended to be.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use publicly available datasets that are commonly used by researchers. And our work mainly focuses on representation learning.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We use publicly available datasets that are commonly used by researchers and we cite the paper of the open code and datasets we used, where the detailed documentation can be found.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4 Experiments section.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

4 Experiments section.

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We use well-known language models BERT-base and RoBERTa-base.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4 Experiments Section and Implementation Details section in Appendix.

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We fix the random seed in all our experiments.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4 Experiments Section and Implementation Details section in Appendix.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.