Representation Degeneration Problem in Prompt-based Models for Natural Language Understanding

Qingyan Zhao[1,2], Ruifang He[1,2,*], Jinpeng Zhang[1,2], Chang Liu[1,2,3], Bo Wang[1,2]
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China
{zhaoqingyan, rfhe, zjpbinary, bo_wang}@tju.edu.cn
[3]aaronl@stu.xjtu.edu.cn

Abstract

Prompt-based fine-tuning (PF), by aligning with the training objective of pre-trained language models (PLMs), has shown improved performance on many few-shot natural language understanding (NLU) benchmarks. However, the word embedding space of PLMs exhibits anisotropy, which is called the representation degeneration problem. In this paper, we explore the self-similarity within the same context and identify the anisotropy of the feature embedding space in PF model. Given that the performance of PF models is dependent on feature embeddings, we inevitably pose the hypothesis: this anisotropy limits the performance of the PF models. Based on our experimental findings, we propose CLMA, a **C**ontrastive **L**earning framework based on the `[MASK]` token and **A**nswers, to alleviate the anisotropy in the embedding space. By combining our proposed counter-intuitive SSD, a **S**upervised **S**ignal based on embedding **D**istance, our approach outperforms mainstream methods on the many NLU benchmarks in the few-shot experimental settings. In subsequent experiments, we analyze the capability of our method to capture deep semantic cues and the impact of the anisotropy in the feature embedding space on the performance of the PF model.

Keywords: anisotropy, prompt-based fine-tuning, contrastive learning

1. Introduction

The prompt-based fine-tuning (PF) method, by aligning with the training objective of pre-trained language models (PLMs) and without introducing additional trainable parameters, reduces the data requirement during fine-tuning compared to methods such as fine-tuning PLMs with a task-specific head (FT) (Kavumba et al., 2022). This enables the PF method to achieve remarkable performance in many few-shot natural language understanding (NLU) tasks (Le Scao and Rush, 2021). PF models convert any natural language processing (NLP) task into a cloze prompt (Petroni et al., 2019) or prefix prompt (Lester et al., 2021) format, aligning with the training objective of masked language models (MLMs) (Devlin et al., 2019) or autoregressive language models (Lewis et al., 2020). Figure 1 illustrates the framework of a PF model for handling natural language inference (NLI) tasks. The PF model utilizes an *MLM* to predict the answer at the masked position. Subsequently, a *verbalizer* is used to map this answer to the corresponding label. The input of the *MLM* is constructed using specific template, which ensures that all *answers* conform to the syntactic structure of the context. In this case, the word embeddings at the masked positions are referred to as feature embeddings (Jiang et al., 2023). Therefore, the performance of PF models is closely related to the feature embeddings (Liu et al., 2023).

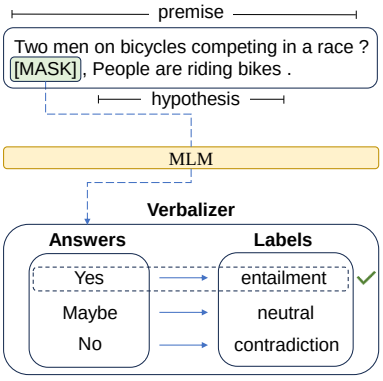

Figure 1: The process of using the PF method to complete the NLI task.

However, the word embeddings outputted by PLMs occupy an anisotropic cone in the vector space, which is known as the representation degeneration problem (Gao et al., 2019; Wang et al., 2020) and limits the performance of language models (Gao et al., 2019; Mu and Viswanath, 2018). Unlike previous methods that analyze the self-similarity of the same token across different contexts and the intra-sentence similarity of tokens at different positions within the same context (Xiao et al., 2023; Cai et al., 2021; Ethayarajh, 2019), we investigate the relationship between the embeddings of the `[MASK]` token and answers at the same position within the same context. We find that within the same context, a remarkably high cosine similarity exists not only between any two an-

*Corresponding author

LREC-COLING 2024, pages 13946–13957

swers but also between the [MASK] token and any individual answer. This indicates that traditional PF methods overlook the alleviation of anisotropy in the feature embedding space. Based on this, we propose a hypothesis: *the anisotropy of the feature embedding space also limits the performance of PF models*.

Combining our experimental findings, we propose a contrastive learning framework, based on the [MASK] token and answers in the same context, that aims to maximize the distance between the [MASK] token and answers, while minimizing the distance between different answers, instead of relying on traditional methods of minimizing intra-class distance and maximizing inter-class distance within the same batch (Xu et al., 2023; Jian et al., 2022; Gao et al., 2021b; Yan et al., 2021). By combining our proposed data augmentation method, we effectively alleviate the anisotropy of the feature embedding space. As far as we know, this is the first method that utilizes a contrastive learning framework to regularize the feature embedding space for PF models.

Additionally, we discover a unique relationship among the aforementioned feature embeddings. Leveraging this particular finding, we believe that the answer with the greatest embedding distance from the [MASK] token within the same context is the one that the model should output, and we use this insight as a counter-intuitive supervised signal. Employing these methods, our model achieves outstanding performance on multiple NLU benchmarks, compared to other mainstream few-shot learning methods and non-fine-tuned large language models (LLMs). Furthermore, when we directly transfer the models trained on MNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015) to the HANS (McCoy et al., 2019) dataset for evaluation, our model maintain the same optimal performance with minimal performance degradation. Additional experimental results indicate that the performance improvement of the PF model is not entirely positively correlated with the alleviation of anisotropy in the feature embedding space.

In summary, our main contributions are as follows:

1. We verify the anisotropy in the feature embedding space of the PF model, leading us to propose our hypothesis: *The anisotropy in the feature embedding space of PF models also limits the performance of PF models*.

2. Based on our hypothesis, we innovatively propose a contrastive learning framework to regulate the feature embedding space in PF models. By incorporating our proposed counter-intuitive supervised signals, our approach not only achieves performance improvements on multiple few-shot NLU benchmarks but also demonstrates effectiveness in capturing deep semantic cues.

3. Through further analysis of the experimental results, we find that although our method mitigates the anisotropy in the feature embedding space while enhancing the performance of the PF model, the performance improvement is not entirely positively correlated with the reduction of the anisotropy.

2. Related Work

2.1. Representation Degeneration

The representation degeneration problem refers to the phenomenon where word embeddings of language models occupy a narrow cone of anisotropy in the vector space (Gao et al., 2019; Ethayarajh, 2019). Xiao et al. (2023); Ethayarajh (2019); Cai et al. (2021) analyze the anisotropy of word representations from various hidden layers of the PLMs by comparing their self-similarity and intra-sentence similarity. Wu et al. (2023) analyze the anisotropy within the same utterance and between different utterances in dialogue models. However, they have not focused on the relationship between different word embeddings at the same position within the same context. Previous works (Li et al., 2020; Wang et al., 2020; Yan et al., 2021; Gao et al., 2021b; Abaskohi et al., 2023; Xu et al., 2023) have attempted to address the issue of representation degeneration, but they have not focused on the problem of representation degeneration in PF models. In this paper, we specifically analyze the self-similarity of feature embeddings in the same context for the PF models and propose a novel contrastive learning framework based on feature embeddings.

2.2. Prompting

Prompt-based fine-tuning has become a new paradigm in NLP (Abaskohi et al., 2023; Gao et al., 2021a; Schick and Schütze, 2021; Liu et al., 2023; Shin et al., 2020; Brown et al., 2020; Petroni et al., 2019). Xu et al. (2023) employ a contrastive learning approach to obtain embeddings for the [MASK] token in task-invariant continuous prompt-tuning however, they still use the [MASK] tokens from other samples in the same batch as positive and negative samples for contrastive learning, without investigating the relationship between the embeddings of [MASK] tokens and answers. In addition, previous research has shown that PF methods exploit dataset-specific superficial cues (Kavumba et al., 2022). Therefore, PF models trained on MNLI or SNLI datasets, tend to exhibit poorer performance when evaluated on NLI datasets like HANS (McCoy et al., 2019) that do

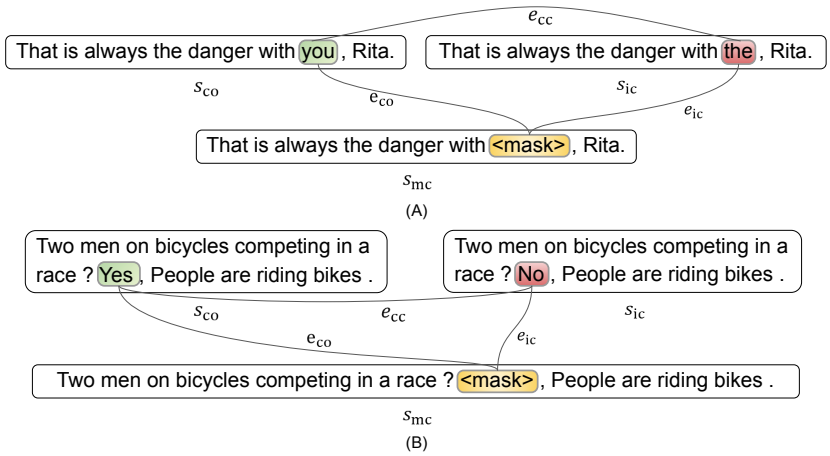

Figure 2: (A) An example of our exploratory experiments conducted in the single-sentence mask filling scenario. (B) An example of our exploratory experiments conducted in the NLI scenario.

not contain effective superficial cues. This indicates that in previous works, the semantic information in the embeddings of PLMs has not been fully utilized.

2.3. Contrastive learning

Recently, contrastive learning has also been widely applied in NLP tasks. By combining certain data augmentation strategies, it has improved the text representation capability by reducing the distance between text representations that have the same semantics in the semantic space. This approach has shown promising results in many low-resource NLP scenarios (Abaskohi et al., 2023; Xu et al., 2023; Jian et al., 2022; Giorgi et al., 2021; Yan et al., 2021; Gao et al., 2021b). Abaskohi et al. (2023); Xu et al. (2023); Jian et al. (2022) combine PF models with contrastive learning framework to handle NLU tasks. However, previous methods mostly relied on intra-class or inter-class contextual embeddings and did not explore the relationship between the `[MASK]` token and answers in the feature embedding space within the PF models.

3. Exploration of the Feature Embedding Space

Unlike previous methods that investigate self-similarity and intra-sentence similarity (Xiao et al., 2023; Ethayarajh, 2019; Cai et al., 2021), we specifically investigate the cosine similarity between word embeddings of different words at the same position in the same context for the PF model. We prepare sentence triplets as shown in the Figure 2, where each triplet consists of a complete and correct sentence s_{co}, a sentence s_{mc} obtained by replacing one token in s_{co} with a `[MASK]` token, and a sentence s_{ic} obtained by replacing the `[MASK]` token in s_{mc} with an incorrect word that does not fit the context. These sentences are then fed into the same MLM to obtain the word embeddings r_{co}, r_{mc} and r_{ic} at the masked positions. Taking r_{co}, r_{mc} as examples:

$$r^i_{\text{co}} = \text{MLM}(s^i_{\text{co}}); \quad r^i_{\text{mc}} = \text{MLM}(s^i_{\text{mc}}), \tag{1}$$

where MLM$(\cdot)$ denotes the function that maps s^i to the word embedding r^i at the masked position in the last layer of MLM. Then we calculate the cosine similarity e_{co}, e_{cc} and e_{ic} between r_{co} and r_{mc}, r_{co} and r_{ic}, r_{mc} and r_{ic}, respectively. In summary, for each dataset, we compute the average values of e_{co}, e_{cc} and e_{ic}, denoted as E_{co}, E_{cc} and E_{ic}. Taking E_{co} as an example:

$$\begin{aligned} e^i_{\text{co}} &= \cos(r^i_{\text{mc}}, r^i_{\text{co}}) \\ E_{\text{co}} &= \frac{1}{k}\sum_{i=1}^{k} e^i_{\text{co}}, \end{aligned} \tag{2}$$

where k denotes the number of samples in the dataset and cos$(\cdot)$ denotes the function for computing cosine similarity.

First, we conducted experiments in a single-sentence mask filling scenario (Figure 2(A)). Specifically, we select every premise and hypothesis sentence from the MNLI-matched and SNLI dev datasets as s_{co} in the aforementioned triplet at first. Then, following the rules mentioned above, we replace a randomly chosen token in s_{co} with a `[MASK]` token to obtain s_{mc}, and then replace the `[MASK]` token in s_{mc} with another randomly chosen token from the vocabulary to obtain s_{ic}. Finally, we calculate cosine similarities for each triplet, resulting in the sets of similarities $(e^1_{\text{co}}, e^2_{\text{co}}, \cdots, e^k_{\text{co}}) \in \mathcal{E}_{\text{co}}$, $(e^1_{\text{cc}}, e^2_{\text{cc}}, \cdots, e^k_{\text{cc}}) \in \mathcal{E}_{\text{cc}}$ and $(e^1_{\text{ic}}, e^2_{\text{ic}}, \cdots, e^k_{\text{ic}}) \in \mathcal{E}_{\text{ic}}$, k denotes the number

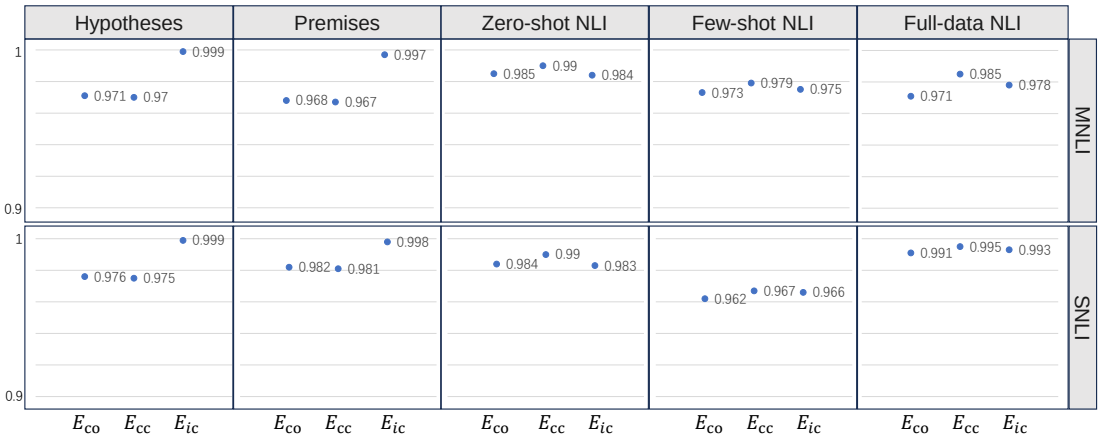

Figure 3: The figure presents the experimental results of constructing triplets using the MNLI-matched and SNLI dev datasets and calculating the average feature embedding similarities within the triplets.

of samples in the dataset. And we compute the average of the sets of similarities $\mathcal{E}_{\text{co}}$, $\mathcal{E}_{\text{cc}}$ and $\mathcal{E}_{\text{ic}}$. The two experimental settings in this case are as follows:

- **Hypotheses**: Constructing triplets based on all hypotheses from the MNLI-matched and SNLI dev datasets.
- **Premises**: Constructing triplets based on all premises from the MNLI-matched and SNLI dev datasets.

The above experiments are based on filling randomly masked words in single sentences using an MLM, which is straightforward for PLMs (Devlin et al., 2019). To better validate our findings, we introduced a more challenging NLI scenario as a comparison (Figure 2(B)). For each premise and hypothesis pair in the MNLI-matched and SNLI dev datasets, we use a template to construct a PF model input of NLI task (Schick and Schütze, 2021; Gao et al., 2021a), which corresponds to the s_{mc} mentioned earlier. Following the same rules, we construct a sentence triplet for comparison. In the triplet, the contextually aligned sentence, denoted as s_{co}, replaces the `[MASK]` token with the correct answer, while the non-contextual sentence, denoted as s_{ic}, replaces the `[MASK]` token with an answer corresponding to one of the other labels. Next, we input the triplet into prompt-based NLI models under three different experimental settings: zero-shot, few-shot, and full data. In this case, we have the following three experimental settings:

- **Zero-shot NLI**: Conducting zero-shot experiments using the PF method on the MNLI-matched and SNLI dev datasets.
- **Few-shot NLI**: Conducting few-shot experiments using the PF method on the MNLI-matched and SNLI dev datasets, the model was trained on 16 samples per class from the MNLI or SNLI train datasets.
- **Full-data NLI**: Conducting full-data experiments using the PF method on the MNLI-matched and SNLI dev datasets, the model was trained on the entire MNLI or SNLI train datasets.

We have made three observations (Figure 3):

- The feature embeddings of PF models exhibit anisotropy across different experimental settings. This is demonstrated by the high cosine similarity of the feature embeddings found in all scenarios.
- In all NLI scenarios, we found that E_{cc} is the highest among the three averages. Therefore, we conclude that compared to the special token `[MASK]`, the embeddings of correct and incorrect answers have a higher cosine similarity.
- In the zero-shot NLI scenario, not only the performance of the model is poor (Schick and Schütze, 2021), but also the mean of similarity $\mathcal{E}_{\text{co}}$ (E_{co}) is greater than the mean of similarity $\mathcal{E}_{\text{ic}}$ (E_{ic}). This is the only case where E_{co} is greater than E_{ic}.

Based on our experimental findings, we propose a contrastive learning framework based on answers and `[MASK]` token to alleviate the anisotropy in the feature embedding space. Among them, we treat all candidate answers as one category and minimize their intra-class distance, while treating the `[MASK]` token as another category and maximizing its inter-class distance from the candidate answers. Additionally, we introduce a counter-intuitive supervised signal based on feature embedding distances, where the answer with the farthest feature embedding distance

from the [MASK] token is determined as the output of the model.

4. Approach

In this section, we present the method we have proposed. Given an NLU dataset $\mathcal{X}$, a prompting template P, and a set of answers $(z_1, z_2, \cdots, z_n) \in \mathcal{Z}$ corresponding to different labels $\mathcal{Y}$, our objective is to first construct sentence n+1-tuples using P on $\mathcal{X}$. Each n+1 tuple consists of n sentences constructed using answers from $\mathcal{Z}$ and one sentence constructed using the [MASK] token. We then regularize the feature embedding space for $\mathcal{Z}$ based on our proposed contrastive learning framework and finally use the feature embedding distance between the [MASK] token and the answers in $\mathcal{Z}$ as the supervised signal for the model. We begin by presenting our approach for regularizing the feature embedding space and the data augmentation strategies. Subsequently, we introduce the counter-intuitive supervised signal based on the feature embedding distance.

4.1. Regularization of the Embedding Space

4.1.1. Contrastive Learning Framework

We propose a **C**ontrastive **L**earning framework based on the [**M**ask] token and **A**nswers (CLMA) to assist in regularizing the feature embedding space. Based on the experimental conclusions mentioned earlier, we argue that all the answers should have relatively closer distances in the feature embedding space because they at least conform to the syntactic structure of the context. On the other hand, we expect the [MASK] token, as a special token, to have relatively farther distances from the answers in the feature embedding space. Specifically, we utilize the method described in § 3, for an n-way NLU task, we can obtain s_{mc} with a [MASK] token and s_{co} with the answer corresponding to the correct label, along with $(s_{\text{ic}}^1, s_{\text{ic}}^2, \cdots, s_{\text{ic}}^{n-1}) \in \mathcal{S}_{\text{ic}}$, composed of answers corresponding to n−1 incorrect labels. Subsequently, we feed the aforementioned n+1 sentences into the same MLM and obtain the feature embeddings, r_{mc}, r_{co}, $(r_{\text{ic}}^1, r_{\text{ic}}^2, \cdots, r_{\text{ic}}^{n-1}) \in \mathcal{R}_{\text{ic}}$ at the masked position in s_{mc}, s_{co} and $\mathcal{S}_{\text{ic}}$, respectively. Here $r_{\text{mc}}, r_{\text{co}} \in \mathbb{R}^{1\times d}$,$\mathcal{R}_{\text{ic}} \in \mathbb{R}^{(n-1)\times d}$ and d refers to the hidden dimension. Finally, we calculate the Euclidean distance d_{co} between r_{mc} and r_{co}, as well as $(d_{\text{cc}}^1, d_{\text{cc}}^2, \cdots, d_{\text{cc}}^{n-1}) \in \mathcal{D}_{\text{cc}}$ between r_{co} and $\mathcal{R}_{\text{ic}}$:

$$\begin{aligned} d_{\text{co}} &= \text{dis}(r_{\text{mc}}, r_{\text{co}}) \\ \mathcal{D}_{\text{cc}} &= (\text{dis}(r_{\text{co}}, r_{\text{ic}}^1), \cdots, \text{dis}(r_{\text{co}}, r_{\text{ic}}^{n-1})). \end{aligned} \tag{3}$$

In the aforementioned equation, dis(·) represents a text distance calculation function of Euclidean distance. Next, we employ the contrastive learning approach by utilizing d_{co} and $\mathcal{D}_{\text{cc}}$ to compute the InfoNCE loss (Kong et al., 2020). This allows us to minimize the distance between $\mathcal{R}_{\text{ic}}$ and r_{co} while simultaneously maximizing the distance between r_{mc} and r_{co}:

$$\mathcal{L}_{\text{cl}} = -\log \frac{\exp(d_{\text{co}}/\tau)}{\sum_1^n \exp(d_i/\tau)}. \tag{4}$$

In this context, n refers to the number of label categories and the total number of feature embedding distances in d_{co} and $\mathcal{D}_{\text{cc}}$.

4.1.2. Data Augmentation

In this section, we introduce a data augmentation strategy specifically designed to maintain the diversity of the anchor sample s_{mc} when constructing them for contrastive learning (Khosla et al., 2020). Specifically, for premises and hypotheses labeled as "entailment", we simply set both sentences as either the premise sentence or the hypothesis sentence and then incorporate them into the template. We do not use data augmentation methods such as Cutoff (Shen et al., 2020) or Token Shuffling (Lee et al., 2020). The reason for this is that we are concerned that these random data augmentation strategies may alter the semantic relationship between the two sentences, thus introducing additional noise to the model. For premises and hypotheses labeled as "non-entailment" (McCoy et al., 2019), we apply the Cutoff method for data augmentation on the premise. Similarly, because any random data augmentation applied to the hypothesis could potentially alter the semantic relationship between the premise and hypothesis. In order to avoid introducing additional noise, we do not apply any data augmentation to the hypotheses, and then incorporate them into the template. This way, we obtain the anchor sample s_{mc}.

4.2. Counter-intuitive Supervised Signals

We also propose a **S**upervised **S**ignal based on feature embedding **D**istance (SSD), which allows the supervised signal to better align with the training objective during the regularization of the embedding space. Based on the experimental conclusions from the previous sections, we believe that the answer, which has a greater embedding distance from the [MASK] token in the same context, is the answer that the model supposed to output. Specifically, for each pair of premise and hypothesis sentences, without performing data augmentation, we directly place them into the template to

generate the input s_{mk} for PF model. Then, following the method described earlier (§ 3), we replace the [MASK] token with the answers corresponding to each category label, and construct input sentences $(s^1_{ans}, s^2_{ans}, \cdots, s^n_{ans}) \in \mathcal{S}_{ans}$ for the same PLM. After obtaining the word embedding r_{mk} for the [MASK] token and $(r^1_{ans}, r^2_{ans}, \cdots, r^n_{ans}) \in \mathcal{R}_{ans}$ for the answers, we directly calculate the embedding distance $(d^1_{ans}, d^2_{ans}, \cdots, d^n_{ans}) \in \mathcal{D}_{ans}$ between r_{mk} and $\mathcal{R}_{ans}$:

$$d_{ans} = \text{Concat}(\text{dis}(r_{mk}, r^1_{ans}) : \text{dis}(r_{mk}, r^n_{ans})). \quad (5)$$

Then the objective can be expressed as following:

$$\mathcal{L}_{sup} = \text{CrossEntropy}(d_{ans}, y), \quad (6)$$

where y denotes the correct label.

4.3. Joint Training

We jointly train the model with the above two objectives $\mathcal{L}_{cl}$ and $\mathcal{L}_{sup}$ on NLU datasets. α is a hyperparameter to balance two objectives (Yan et al., 2021):

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{cl} + (1-\alpha)\mathcal{L}_{sup}. \quad (7)$$

Algorithm 1: Our Method

MaxStep = The number of training steps;
Sample: Randomly sampling function;
Train_Set: Training set;
DA: Data augmentation;
LM: Language model;
CL: Contrastive loss;
DIS: Euclidean distance;
SUP: Cross Entropy loss;
for i *in MaxStep* **do**
  $s_{mk}, \mathcal{S}_{ans}, y = \textit{Sample}(\textit{Train\_Set})$;
  $s_{mc}, s_{co}, \mathcal{S}_{ic} = \textit{DA}(s_{mk}, \mathcal{S}_{ans})$;
  // *CLMA*
  $r_{mc}, r_{co}, \mathcal{R}_{ic} = \textit{LM}(s_{mc}, s_{co}, \mathcal{S}_{ic})$;
  $\mathcal{L}_{cl} = \textit{CL}(r_{mc}, r_{co}, \mathcal{R}_{ic})$;
  // *SSD*
  $r_{mk}, \mathcal{R}_{ans} = \textit{LM}(s_{mk}, \mathcal{S}_{ans})$;
  $\mathcal{L}_{sup} = \textit{SUP}(\textit{DIS}(r_{mk}, \mathcal{R}_{ans}), y)$;
  // *Joint Training*
  $\mathcal{L}_{total} = \alpha\mathcal{L}_{cl} + (1-\alpha)\mathcal{L}_{sup}$;
  $\mathcal{L}_{total}.backward()$;
  $optimizer.step()$;
end

5. Experiments

To validate the effectiveness of our method 1, we conduct few-shot NLU experiments.

5.1. Setups

5.1.1. Dataset

Following previous works (Xu et al., 2023; Kavumba et al., 2022; Gao et al., 2021a; Utama et al., 2021; McCoy et al., 2019), we validate our method on multiple NLU datasets, including MRPC (Dolan and Brockett, 2005), QQP [1], MNLI, SNLI, HANS, QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2006). Among them, MRPC and QQP belong to short text matching (STM) tasks, while MNLI, SNLI, HANS, QNLI, and RTE are related to NLI tasks.

Following Xu et al. (2023), for each dataset, we maintain 16 samples per label in the training set. All experimental results are derived from the means of five different training sets.

5.1.2. Baselines

To demonstrate our performance, we consider the following methods as strong baselines:

- **FT**: fine-tuning based on the task-specific head.
- **PF** (Liu et al., 2023): prompt-based fine-tuning using manual templates and answers.
- **LM-BFF+SupCon** (Jian et al., 2022): it is a contrastive learning framework that combines contrastive loss with the standard masked language modeling loss in prompt-based few-shot learners.
- **CP-Tuning** (Xu et al., 2023): it is an end-to-end contrastive prompt tuning framework,without any manual engineering of task-specific prompts and verbalizers.
- **PF+CLMA**: it employs PF to replace the SSD that we proposed. The aim is to validate the effectiveness of the supervision signal we introduced.
- **Chatglm2-6B**[2]: it is an open-source bilingual Chinese-English dialogue large language model based on GLM (Du et al., 2022).
- **InternLM-7B**[3]: it is a multilingual language model with progressively enhanced capabilities.

In the aforementioned prompt-based fine-tuning scenarios, we use the same templates and answer keys as Gao et al. (2021a) (Table 2).

[1] https://www.quora.com/q/quoradata/
[2] https://github.com/THUDM/ChatGLM2-6B
[3] https://github.com/InternLM/InternLM

Backbone	Method	Few-Shot STM (acc/F1)		Few-Shot NLI (acc)						Migrate to HANS (acc)	
		MRPC	QQP	$MNLI_m$	$MNLI_{mm}$	SNLI	HANS	QNLI	RTE	MNLI-HANS	SNLI-HANS
ALBERT	FT	62.03 / 70.69	67.36 / 73.31	58.47	58.85	62.48	66.25	54.57	50.90	50.69	54.27
	PF	65.80 / 75.82	73.24 / 78.23	58.24	59.75	71.71	69.24	59.93	59.93	51.01	60.27
	CP-Tuning†	63.52 / -	71.05 / -	-	-	-	-	62.02	61.92	-	-
	PF+CLMA	**66.90** / **77.26**	**75.1** / **80.12**	57.94	59.74	74.79	67.39	61.08	59.57	51.22	58.91
	OURS	65.51 / 74.28	74.74 / 79.58	**65.94**	**63.94**	**78.96**	**71.15**	**62.58**	**64.26**	**52.52**	**64.72**
RoBERTa	FT	69.86 / 78.08	60.51 / 63.50	57.52	59.15	72.57	64.06	70.84	69.31	50.00	50.15
	PF	71.36 / 79.70	67.21 / 70.86	74.16	75.06	78.59	64.48	71.43	68.95	53.16	56.76
	LM-BFF+SupCon†	- / 77.80	**74.00** / -	72.40	74.20	79.60	-	71.10	71.80	-	-
	CP-Tuning†	72.60 / -	73.56 / -	-	-	-	-	69.22	67.22	-	-
	PF+CLMA	70.72 / 79.05	69.53/ 72.35	69.88	70.47	80.30	73.06	69.92	71.48	50.10	52.32
	OURS	**74.90** / **82.92**	72.55 / **75.59**	**75.45**	**75.81**	**81.41**	**74.53**	**72.82**	**72.20**	**66.39**	**58.81**
Chatglm2-6B‡		73.82 / 80.39	75.33 / 82.72	64.79	66.26	72.77	66.43	66.89	76.17	-	-
InternLM-7B‡		67.94 / 73.58	**80.34** / **85.59**	65.91	66.73	72.05	63.41	62.47	**80.51**	-	-

Table 1: The table displays our experimental results. The left half of the table presents the few-shot experimental results on STM and NLI tasks, while the right half shows the performance of models trained on MNLI or SNLI and directly transferred to HANS for evaluation. Methods with † indicate that we directly report the scores from the corresponding paper. Method with ‡ indicate a zero-shot experimental setup.

Tasks	Template	Answer Keys
STM	$<S_1>$ `[MASK]`, $<S_2>$	Yes, No
NLI	$<S_1>$? `[MASK]`, $<S_2>$	Yes, Maybe, No

Table 2: The templates and answer keys used in our experiments. Among them, "Maybe" is used only in three-way NLI tasks.

5.1.3. Hyperparameters

In the experiment, the ratio of feature cutoff is set to 0.2 , as suggested in Shen et al. (2020), and the batch size is set to 16 in most of our experiments. We use the RMSprop optimizer and set the learning rate and the ratio of warm-up to 5e-6 and 10% respectively. We evaluate the impact of temperature on the performance of our model and select 0.1 as the temperature in most of our experiments. Finally, we train the model for 1000 steps and evaluate the model every 100 steps during training.

5.2. Results

5.2.1. Few-Shot NLU

We compare the performance of models using RoBERTa-large (Liu et al., 2019) and ALBERT-xxlarge (Lan et al., 2020) as PLMs in the few-shot experimental setting. The experimental results are shown in Table 1. In our experiments, the prompt-based models outperformed the FT models on most datasets, demonstrating the effectiveness of the PF method in few-shot experiments. Furthermore, large language models, with the advantages of a greater number of model parameters and more training data (Du et al., 2022), possess stronger logical reasoning capabilities. Despite being in a zero-shot experimental setting, their performance on QQP and RTE datasets still surpasses the few-shot experimental results of mainstream methods. Lastly, compared to mainstream methods in a few-shot experimental setting and large language models in a zero-shot experimental setting, our method achieves optimal results on the vast majority of datasets. Additionally, the comparative results with the PF+CLMA method prove the necessity of the SSD we proposed.

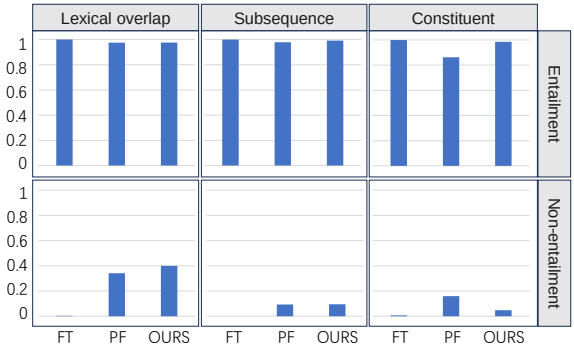

Figure 4: The performance (acc) of evaluating the FT model, PF model, and our model on the HANS dataset under different labels and syntactic heuristics. These models are all trained on the SNLI dataset under a few-shot experimental setting.

5.2.2. Migrate to HANS dataset

To verify the robustness of our model, we transfer the best-performing models trained on MNLI and SNLI directly to the HANS test dataset to evaluate their ability to recognize deep semantic clues (Kavumba et al., 2022). The experimental results are shown in the right half of Table 1. Our method outperforms other approaches when transferred to the HANS dataset. Specifically, compared to the PF method, our approach achieves an average performance improvement of 5.31%. This demonstrates that our method enhances the ability of tra-

Method	MNLI_m	MNLI_{mm}	SNLI	MNLI-HANS	SNLI-HANS
Full Implement.	**75.45**	**75.81**	**81.41**	**66.39**	**58.81**
w/o. CLMA	73.21	74.53	80.08	64.57	55.83
w/o. SSD	64.69	66.80	53.13	51.07	51.91

Table 3: Ablation study regarding model performance. ‘‘Full Implement.” refers to the full implementation of our method.

	MNLI			SNLI		
DataSet	E_{co}	E_{cc}	E_{ic}	E_{co}	E_{cc}	E_{ic}
Full Implement.	0.8930	0.9409	0.9235	0.8692	0.9489	0.9200
w/o. CLMA	0.8860	0.8858	0.9390	0.8787	0.8646	0.9290
w/o. SSD	0.8511	0.9397	0.8559	0.9290	0.9999	0.9017

Table 4: Ablation study on alleviating anisotropy in the feature embedding space. ‘‘Full Implement.” refers to the full implementation of our method.

ditional PF models to grasp deep semantic cues.

In Figure 4, we present the transfer performance of the FT model, PF model, and our method on the HANS dataset. We evaluate the performance of the three models considering various labels and syntactic heuristics. Consistent with previous studies (McCoy et al., 2019), we observe that all models trained on the SNLI dataset exhibit superior performance on ‘‘entailment” labeled data and underperform on ‘‘non-entailment” labeled data in the HANS dataset evaluation. Compared to the other two methods, our approach achieves comparable performance on the ‘‘entailment” label while also achieving relatively good performance on the ‘‘non-entailment” label. It only performs slightly worse than the PF model under the ‘‘Constituent” syntactic heuristic.

5.2.3. The Regularized Feature Embedding Space

We conduct a similar exploration as described in (§ 3), investigating the regularized feature embedding space using our proposed method. The experimental results shown in Table 4 demonstrate that our method effectively reduces the similarity between answers in the feature embedding space, as well as the similarity between the `[MASK]` token and answers, thus alleviating the anisotropy of the feature embedding space.

5.3. Ablation Study

We conduct an ablation study to investigate the characteristics of the main components in our method, including CLMA and SSD. Table 3 reports the accuracy on MNLI-match, MNLI-mismatch, SNLI, and the corresponding evaluation accuracy on the HANS task. From the table, we can observe a performance drop when removing these components. Specifically, when only SSD are retained, the model trained on MNLI and SNLI exhibits a substantial decrease in evaluation performance on HANS, demonstrating the effectiveness of our proposed contrastive learning framework in capturing deep semantic clues. Similarly, when SSD are removed, the performance of the model also declines, validating the effectiveness of our proposed feature embedding distance-based supervised signal.

Table 4 displays the roles of CLMA and SSD in alleviating the anisotropy of the feature embedding space. Among them, SSD can effectively reduce the cosine similarity E_{cc}, while CLMA can effectively decrease the cosine similarities E_{co} and E_{ic}.

Furthermore, by analyzing the results of the ablation studies, we find that when only CLMA or SSD is retained, anisotropy in the feature embedding space of the model might even decrease, but the performance of the model worsens. This indicates that a reduction in anisotropy in the feature embedding space is not always directly correlated with an improvement in model performance.

6. Conclusions

We begin by exploring the connection between answers and the `[MASK]` token within the feature embedding space of the PF model, leading us to formulate our hypothesis. Based on our experimental findings, we introduce a contrastive learning framework that utilizes answers and the `[MASK]` token to alleviate anisotropy in the feature embedding space. Subsequently, we propose a novel, counter-intuitive, supervised signal that hinges on the distance within the feature embedding space. Our experiments demonstrate the effectiveness

of our approach for few-shot NLU tasks, particularly in capturing deep semantic clues. Finally, further experimental results verify that, although our model enhances performance while alleviating anisotropy in the feature embedding space, the reduction of anisotropy and the improvement of model performance are not positively correlated.

7. Limitations

We have listed some limitations: (a) Our method requires multiple forward passes with PLM at each training step to obtain the feature embeddings. This to some extent leads to a loss in model runtime speed, which is an area for future optimization. (b) In our experiments, we followed previous related work and focused solely on the NLU task. We did not evaluate the performance of our method on tasks such as question-answering, text classification. This is also a subject for future research.

8. Acknowledgement

This work was supported by National Natural Science Foundation of China (62376192, 62376188).

9. References

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, pages 1877–1901.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. 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 320–335.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

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

models. In *International Conference on Learning Representations*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. SemEval-2018 task 12: The argument reasoning comprehension task. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 763–772.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2022. Contrastive learning for prompt-based few-shot language learners. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5577–5587.

Weisen Jiang, Yu Zhang, and James T. Kwok. 2023. Effective structured prompting by meta-learning and representative verbalizer. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15186–15199.

Pride Kavumba, Ryo Takahashi, and Yusuke Oda. 2022. Are prompt-based models clueless? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2333–2352.

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*, pages 18661–18673.

Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636.

Haejun Lee, Drew A. Hudson, Kangwook Lee, and Christopher D. Manning. 2020. SLM: Learning a discourse language representation with sentence unshuffling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1551–1562.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35.

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. abs/1907.11692.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. abs/2009.13818.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235.

Prasetya Utama, Nafise Sadat Moosavi, Victor Sanh, and Iryna Gurevych. 2021. Avoiding inference heuristics in few-shot prompt-based finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9074.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

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

Han Wu, Haochen Tan, Mingjie Zhan, Gangming Zhao, Shaoqing Lu, Ding Liang, and Linqi Song. 2023. Learning locality and isotropy in dialogue modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Chenghao Xiao, Yang Long, and Noura Al Moubayed. 2023. On isotropy, contextualization and learning dynamics of contrastive-based sentence representation learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12266–12283.

Ziyun Xu, Chengyu Wang, Minghui Qiu, Fuli Luo, Runxin Xu, Songfang Huang, and Jun Huang. 2023. Making pre-trained language models end-to-end few-shot learners with contrastive prompt tuning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*, pages 438–446.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Con-

SERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075.