Manifold-Based Verbalizer Space Re-embedding for Tuning-Free Prompt-Based Classification

Haochun Wang, Sendong Zhao*, Chi Liu, Nuwa Xi, Muzhen Cai, Bing Qin, Ting Liu

Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
{hcwang, sdzhao}@ir.hit.edu.cn

Abstract

Prompt-based classification adapts tasks to a cloze question format utilizing the [MASK] token and the filled tokens are then mapped to labels through pre-defined verbalizers. Recent studies have explored the use of verbalizer embeddings to reduce labor in this process. However, all existing studies require a tuning process for either the pre-trained models or additional trainable embeddings. Meanwhile, the distance between high-dimensional verbalizer embeddings should not be measured by Euclidean distance due to the potential for non-linear manifolds in the representation space. In this study, we propose a tuning-free manifold-based space re-embedding method called **L**ocally **L**inear **E**mbedding with **I**ntra-class **N**eighborhood **C**onstraint (**LLE-INC**) for verbalizer embeddings, which preserves local properties within the same class as guidance for classification. Experimental results indicate that even *without tuning any parameters*, our LLE-INC is on par with automated verbalizers with parameter tuning. And with the parameter updating, our approach further enhances prompt-based tuning by up to 3.2%. Furthermore, experiments with the LLaMA-7B, 13B and 65B indicate that LLE-INC is an efficient tuning-free classification approach for the hyper-scale language models.

Introduction

Large language models have seen remarkable success in natural language processing (NLP) with pre-training on vast amounts of unlabeled data via masked language model (MLM) (Devlin et al. 2019; Liu et al. 2019) and fine-tuning for the downstream tasks (Howard and Ruder 2018). Despite these advancements, the above paradigm necessitates a substantial amount of labeled training data, which poses challenges when attempting to train an additional classification layer during fine-tuning with limited data (Liu et al. 2021a).

Prompt-based tuning is a method that addresses this issue by converting the task into a cloze question (Schick and Schütze 2021a), where the pre-trained models (PTMs) are prompted to fill in the [MASK] token with a suitable token from the vocabulary list, similar to the pre-training task.

Only a limited number of tokens from the vocabulary list are chosen as *verbalizers*, which map each token to its corresponding class, as illustrated in Figure 1. Furthermore, variations in the verbalizer designs for a task can significantly impact the performance of prompt-based tuning (Gao, Fisch, and Chen 2021). As shown in Figure 1, recent studies on prompt-based tuning have employed verbalizers in three distinct ways: manually selected token verbalizers (ManualVerb) by domain experts empirically (Schick and Schütze 2021a), automatically generated token verbalizers (GenVerb) through gradient-based searching or language model (Shin et al. 2020; Gao, Fisch, and Chen 2021), and embedded verbalizers (EmbedVerb) through optimizing trainable embeddings (Zhang et al. 2022b; Hambardzumyan, Khachatrian, and May 2021; Cui et al. 2022).

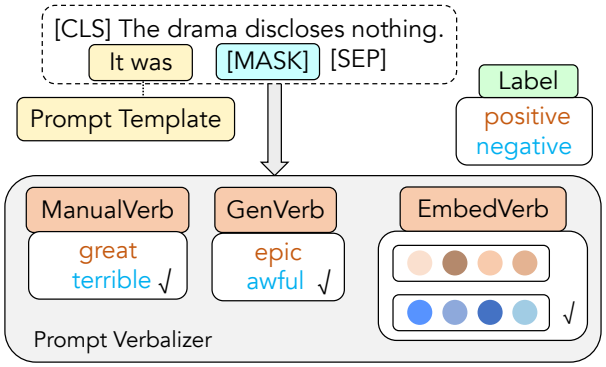

Figure 1: Illustration of three typical verbalizer design methods for prompt-based tuning.

There are two remaining challenges for the prompt-based classification: (1) All the above three kinds of verbalizers require *updating the parameters* either in the PTMs or the training of extra embeddings, which requires great computational resources, especially for the large language models (LLMs). (2) The *potential manifold* comprising the verbalizer embeddings distributed in the high-dimensional space has not been taken into account and the intra-class neighborhood relationship has been ignored. While a wide range of applications treat the observed space in the PTM as a high-dimensional Euclidean space and utilize Euclidean distance to measure the distance between different vectors (Yu et al. 2021; Gao, Yao, and Chen 2021), this may not be as appropriate for the verbalizer embedding on a manifold which is

*Corresponding author

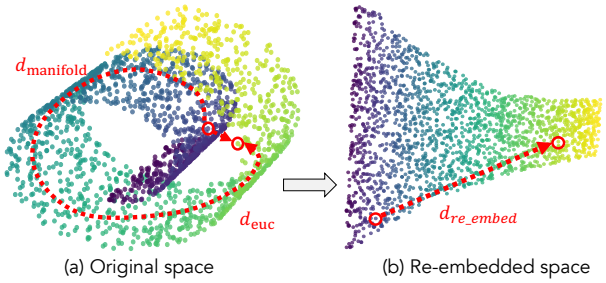

Figure 2: A sketch map for the space re-embedding. d_{euc} is the Euclidean distance in the original space. $d_{manifold}$ is the distance along the manifold shape in the original space. d_{re_embed} is the Euclidean distance in the re-embedded space.

a topological space that only resembles Euclidean space locally (Lee 2010). For instance, in Figure 2, given two 3-D data points marked with red circles, the Euclidean distance between them is d_{euc} but it is clear that when considering the manifold, the distance along the roll $d_{manifold}$ can more accurately depict the relationship between the points, with $d_{euc} \ll d_{manifold}$. Similarly, verbalizer embeddings with even higher dimensions can be also distributed on highly distorted manifolds. Manifold learning can estimate the distance between nearby points with the local neighbors and that between distant points with multi-hop neighbors along the manifold shape, which helps to alleviate this issue.

In this study, we propose manifold-based space re-embedding for tuning-free prompt-based classification. Motivated by the Locally Linear Embedding (Roweis and Saul 2000), we introduce the Locally Linear Embedding with Intra-class Neighborhood Constraint (LLE-INC) for the calibration of [MASK] embeddings in the verbalizer space. We posit that the embeddings of the [MASK] token from the instances that share the same class should be close to one another on a particular manifold. Thus, the LLE-INC aims to reconstruct the representation space into a new Euclidean space, where the linear relationship between data points in the same class is consistent with that in the original space. The recovered Euclidean metric space can improve the distance metrics on high-dimensional verbalizer embeddings, and thus empower the prompt-based PTMs without any tuning on the parameters in the PTMs or the addition of new trainable parameters. The above tuning-free paradigm also explores the potential of LLMs for classification tasks with labeled instances. [1]

In summary, our contributions are as follows:

- We address a significant challenge in the use of embedded verbalizers for prompt-based classification, specifically the potential for Euclidean distance metrics in the original verbalizer space to ignore the existence of manifolds in high-dimensional space.
- We propose the tuning-free LLE-INC method to re-embed the verbalizer space into a new, recovered space, leveraging the intra-class neighborhood relationship in the few-shot datasets. We evaluate the LLE-INC on 10 benchmark datasets. Without tuning any parameters, our approach can produce results that are on par with or superior to baselines. Moreover, when combined with the tuning of PTMs with contrastive learning, our approach outperforms the baselines consistently.
- We explore a novel and efficient strategy of leveraging the output embeddings of PTMs with no parameter updating for tuning-free applications for LLMs.

[1]We release our code at https://github.com/SCIR-HI/LLE-INC.

Related Works

Prompt-based Tuning PTMs have demonstrated effectiveness in various NLP tasks recently (Peters et al. 2018; Devlin et al. 2019; Liu et al. 2019). However, in few-shot settings, fine-tuning PTMs with the additional task-specific linear layer can be challenging. To address this issue, prompt-based tuning has been proposed to adapt downstream tasks to the pre-training paradigm by converting them into a cloze-question format using a prompt template and prompt verbalizer (Schick and Schütze 2021a; Gao, Fisch, and Chen 2021; Du et al. 2023), as illustrated in Figure 1. The PTMs are expected to predict these verbalizers for the [MASK] token, which are then mapped to the corresponding labels. Early attempts used a prompt template and verbalizer selected manually through domain knowledge (Schick and Schütze 2021a; Wang et al. 2022b). Subsequently, Gao et al. (Gao, Fisch, and Chen 2021) and Liu et al. (Liu et al. 2021b) proposed methods for searching for the optimal prompt. Additionally, some works have focused on continuous prompt engineering, which involves freezing the parameters of PTMs and only updating the prompt embedding. Lester et al. (Lester, Al-Rfou, and Constant 2021) and Liu et al. (Liu et al. 2023) replaced the prompt template with trainable embeddings and optimized the prompt embeddings with extra training steps. Li et al. (Li and Liang 2021) applied the continuous prompt to generation tasks. In this study, we focus on the discrete prompt template in natural language which is feasible with the tuning-free setting.

Prompt Verbalizer Engineering Prompt verbalizers map the outputs of the Masked Language Model (MLM) task of PTMs to actual labels (Liu et al. 2021a), and the settings for these verbalizers can significantly impact model performance (Gao, Fisch, and Chen 2021). Schick et al. (Schick and Schütze 2021a,b) used manually designed, task-specific prompt verbalizers, which require domain expertise and a vast amount of time to obtain the optimal verbalizers. To address this problem, Gao et al. (Gao, Fisch, and Chen 2021) and Shin et al. (Shin et al. 2020) proposed generating prompt verbalizers from the vocabulary list by maximizing the conditional probability. However, this approach may result in generated verbalizers that are not coherent in the context (Shin et al. 2020). Hu et al. (Hu et al. 2022) expanded the candidate verbalizer list with tokens that share semantic similarity via external knowledge bases. Recently, Cui et al. (Cui et al. 2022) proposed prototypical verbalizer embeddings, which are learned class prototypes based on training instances. It can achieve state-of-the-art performance for automating verbalizer design, but it is still less effective than elaborate manual verbalizers.

Manifold Learning As data in high-dimensional space can be distributed on a specific non-linear manifold, it can be unreasonable to measure differences using Euclidean distance. To address this problem, manifold learning transforms the data in the original space into a new space, based on the assumption that local manifold space is homeomorphic to Euclidean space Roweis et al. (Roweis and Saul 2000) proposed the locally linear embedding to accomplish non-linear dimension reduction via preserving local properties of high dimensional data in the space reconstruction to exploit local symmetries. Inspired by this, Hasan et al. (Hasan and Curry 2017) re-embedded word embeddings by converting pre-trained vectors to a new Euclidean space with word frequency ranking. Chu et al. (Yonghe et al. 2019) proposed a dynamic word selection method to address the singularity problem in the matrix. Wang et al. (Wang et al. 2022a) integrated manifold-based geometric structures and refined sentence embeddings. In this study, we apply manifold learning to prompt learning and leverage the intra-neighborhood relationship in space re-embedding.

Method

Continuous Verbalizer Embedding

In the previous section, we presented an overview of the general process of prompt-based tuning with discrete verbalizers. Manually elaborating verbalizers, particularly for multi-class classification tasks, can be a time-consuming and labor-intensive process. Inspired by (Jiang et al. 2022; Cui et al. 2022), we utilize the [MASK] token embeddings as the representation of instances and classify test instances based on the embedding distance.

$$h_{\texttt{[MASK]}} = h_{\text{instance}} \tag{1}$$

Since the Euclidean distance between embeddings may not be able to capture the potential manifolds in the embedding space, in the following section, we describe how we adopt the method of manifold-based space re-embedding to re-embed the representation of verbalizers with the prompt-based tuning of PTMs.

Manifold-based Re-embedding

Our manifold-based re-embedding methodology is illustrated in Figure 3. We begin with the output embeddings of the [MASK] token in the original space from the PTM and proceed through the following steps to obtain the re-embedded representation. In step 1, the distributed representations of [MASK] tokens are gathered from all classes in the training set. In step 2, we fit the manifold learning model of **L**ocally **L**inear **E**mbedding with **I**ntra-class **N**eighborhood **C**onstraint (**LLE-INC**) to the collected [MASK] embeddings, allowing for the re-embedding of representation from the original space into a new one. In step 3, the fitted model transforms the [MASK] embeddings in the test instances into the re-embedded space. Finally, in step 4, we apply the k-nearest neighbors (kNN) algorithm to the [MASK] embeddings in test instances for the classification tasks. For easier understanding, a table outlining the notation for the variables used in this study can be found in Appendix .

Step 1. In the few-shot scenario, we have a N-way k-shot setting for the training set. The training samples are provided as inputs to the PTM wrapped with prompts. We collect the output hidden state representation of [MASK] tokens $h_{ij},\ i \in 1, 2, ..., N; j \in 1, 2, ..., k$ as the embeddings in the original PTM space, which is in high dimension.

Step 2. Since the metric based on Euclidean distance is unaware of the possible manifold in the original space, we leverage the original embeddings in Step 1 to fit a manifold-based model. Locally Linear Embedding (LLE) (Roweis and Saul 2000) assumes that the local space is homeomorphic to the Euclidean space and a data point is a linear combination of its neighboring points. Thus, a data sample x_i can be reconstructed by the linear combination of its K-nearest neighbor points.

$$x_i = w_{ij}x_j + w_{ik}x_k + w_{il}x_l + ... \tag{2}$$

It is expected that the linear relationship in Formula 2 with the neighbors can be preserved following reconstruction. However, due to the limited data scale, K-nearest neighbors may not accurately represent the shape of the manifold in the high-dimensional space, as the training instances may be distributed sparsely in the original space. Consequently, we assume that the [MASK] embeddings of the instances within the same class should be situated proximally on a specific manifold in the original space and that the relationship between an arbitrary embedding and its intra-class neighbors should be maintained through the verbalizer space reconstruction. With this assumption, a Locally Linear Embedding with Intra-class Neighborhood Constraint (LLE-INC) model is proposed in order to preserve the linear relationship between intra-class neighbors.

The weight coefficient w_{ijm} is determined by locating the c-nearest neighbors of each [MASK] embedding h_{ij}, that are h_{ijm}, which share the same class with h_{ij} and minimizing the following error function. The sum-to-one constraint in Formula 3, which reflects the intrinsic geometric relationship with the corresponding neighbors (Saul and Roweis 2000), ensures that embeddings are transformation-invariant.

$$\min \sum_{i=1}^{N}\sum_{j=1}^{k}\left\| h_{ij} - \sum_{m=1}^{c} w_{ijm}h_{ijm} \right\|^2 \tag{3}$$

$$\text{s.t.} \sum_{m=1}^{c} w_{ijm} = 1$$

Since $\sum_{m=1}^{c} w_{ijm} = 1$, we have $\mathbf{w}_{ij}\mathbf{I} = 1$, where $\mathbf{I} \in \mathcal{R}^{c\times 1}$ is an all-ones column vector. The objective function can be written in the format of vector, where $\mathbf{h}_{ij} \in \mathcal{R}^{1\times d}$, $\mathbf{w}_{ij} \in \mathcal{R}^{1\times c}$, $\mathbf{H}_{ij} \in \mathcal{R}^{c\times d}$.

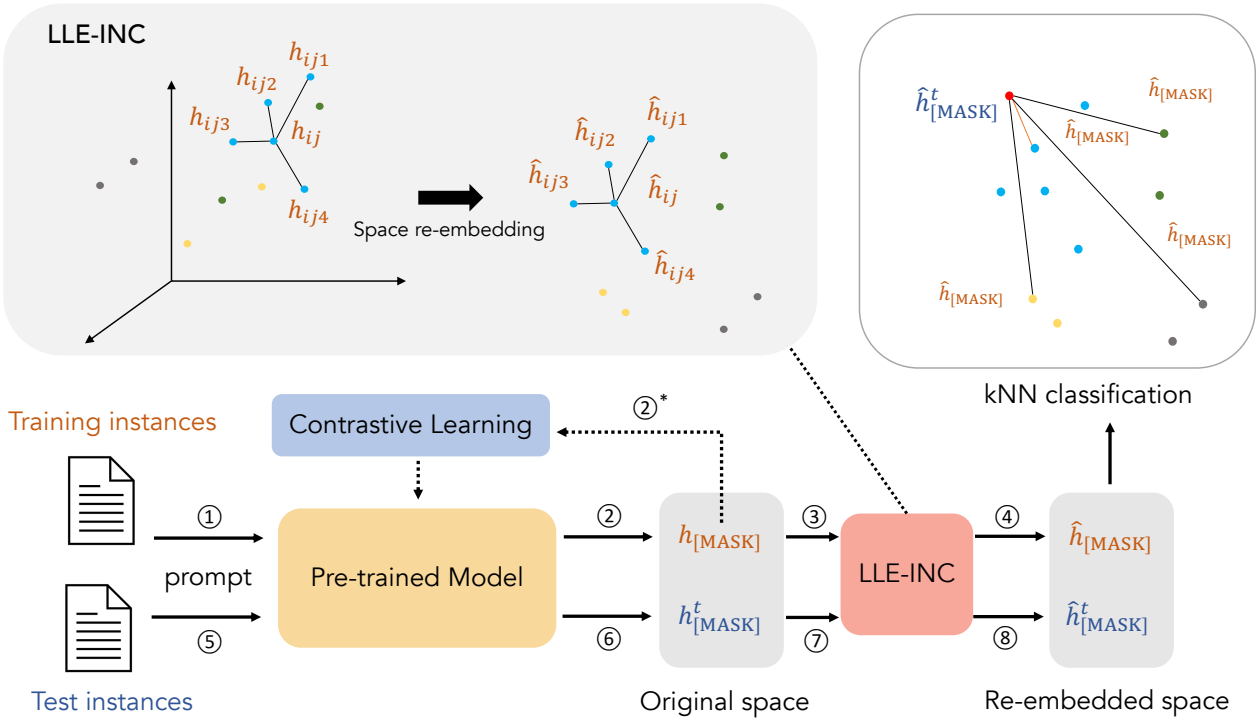

Figure 3: Manifold-based verbalizer space re-embedding for prompt-based tuning. LLE-INC re-embeds the representation space based on the intra-neighbor constraint within the training instances and a kNN classifier makes predictions with the re-embedded representation. The contrastive learning (②*) is a supplementary module and is not essential.

$$
\begin{aligned}
f &= \left\| h_{ij} - \sum_{m=1}^{c} w_{ijm} h_{ijm} \right\|^2 \\
&= \left\| \mathbf{h}_{ij} - \mathbf{w}_{ij}\mathbf{H}_{ij} \right\|^2 \\
&= \left\| \mathbf{w}_{ij}\mathbf{I}\mathbf{h}_{ij} - \mathbf{w}_{ij}\mathbf{H}_{ij} \right\|^2 \\
&= \left\| \mathbf{w}_{ij}(\mathbf{I}\mathbf{h}_{ij} - \mathbf{H}_{ij}) \right\|^2 \\
&= \mathbf{w}_{ij}(\mathbf{I}\mathbf{h}_{ij} - \mathbf{H}_{ij})(\mathbf{I}\mathbf{h}_{ij} - \mathbf{H}_{ij})^T \mathbf{w}_{ij}^T
\end{aligned} \tag{4}
$$

With the Lagrangian multiplier, we can obtain the optimal $\mathbf{w}_{ij}$ for the linear combination.

$$
\mathbf{w}_{ij} = \frac{\mathbf{I}^T((\mathbf{I}\mathbf{h}_{ij} - \mathbf{H}_{ij})(\mathbf{I}\mathbf{h}_{ij} - \mathbf{H}_{ij})^T)^{-1}}{\mathbf{I}^T((\mathbf{I}\mathbf{h}_{ij} - \mathbf{H}_{ij})(\mathbf{I}\mathbf{h}_{ij} - \mathbf{H}_{ij})^T)^{-1}\mathbf{I}} \tag{5}
$$

Afterward, the low-dimensional data can be reconstructed utilizing the weight $\mathbf{w}_{ij}$, and the re-embedded data points should maintain the same relationship with the c-nearest neighbors that possess the same label. Then, the reconstruction error function is as follows, where $\hat{h}_{ij}$, $\hat{h}_{ijm}$ is the re-embedded representation for h_{ij}, h_{ijm} in the new space.

$$
\arg\min_{\hat{h}} \sum_{i=1}^{N} \sum_{j=1}^{k} \left\| \hat{h}_{ij} - \sum_{m=1}^{c} w_{ijm} \hat{h}_{ijm} \right\|^2 \tag{6}
$$

Similar to the above process,

$$
\begin{aligned}
f_{re} &= \left\| \hat{h}_{ij} - \sum_{m=1}^{c} w_{ijm} \hat{h}_{ijm} \right\|^2 \\
&= \left\| \hat{\mathbf{h}}_{ij} - \mathbf{w}_{ij}\hat{\mathbf{H}}_{ij} \right\|^2
\end{aligned} \tag{7}
$$

Let $\hat{\mathbf{H}} = (\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, ..., \hat{\mathbf{h}}_{N\times k}) \in \mathcal{R}^{d'\times(N\times k)}$, where $\hat{\mathbf{h}}_i \in \mathcal{R}^{d'\times 1}$ and d' is the dimension of re-embedded space. $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{N\times k})^T \in \mathcal{R}^{d'\times(N\times k)}$ where $\mathbf{w}_i \in \mathcal{R}^{d'\times 1}$. Let $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$, where $\mathbf{I}$ is the identity matrix for brevity and Formula 6 can be rewritten as

$$
\min_{\hat{\mathbf{H}}} \mathrm{tr}(\hat{\mathbf{H}}\mathbf{M}\hat{\mathbf{H}}^T) \quad \text{s.t. } \hat{\mathbf{H}}\hat{\mathbf{H}}^T = \mathbf{I} \tag{8}
$$

Formula 8 can be solved by eigenvalue decomposition, and the matrix of eigenvectors corresponding to the smallest d' eigenvalues is $\hat{\mathbf{H}}^T$, which comprises the re-embedded representation.

Step 3. During the inference for test instances, the representation of the [MASK] token h_t is transformed into the new verbalizer space with the same dimension above. With Step 2, the weights for the linear combination are constructed with c-nearest neighbors of $h_{tm}, m \in 1, 2, ..., c$ from the training instances in the original space.

$$\min \left\| h_t - \sum_{m=1}^{c} w_{tm} h_{tm} \right\|^2 \tag{9}$$

Then, the weights are employed to combine the re-embedded representation of the corresponding neighbors in Step 2 to transform h_t into $\hat{h}_t$.

$$\hat{h}_t = \sum_{m=1}^{c} w_{tm} \hat{h}_{tm} \tag{10}$$

Step 4. The procedure described above results in the representation of [MASK] tokens in the re-embedded space for both the training instances and test instances. The classification of a given test instance $\hat{h}_t$ is determined through its e-nearest neighbors $\hat{h}_l^t, l \in 1, 2, ..., e$ with the cosine distance from the training instances distributed in the re-embedded space.

$$d(\hat{h}_t, \hat{h}_l^t) = \frac{\hat{h}_t \cdot \hat{h}_l^t}{\left\|\hat{h}_t\right\| \cdot \left\|\hat{h}_l^t\right\|} \tag{11}$$

$$P(y_{\hat{h}_t} = n | X = \hat{h}_t) = \frac{1}{e} \sum_{l=1}^{e} Ind(y_{\hat{h}_l^t} = n) \tag{12}$$

where $Ind(\dots)$ represents an indicator function.

Parameter Updating

Up until this point, no parameter has been tuned throughout the re-embedding process described above, thus avoiding the consumption of computational resources and the storage for the tuned models. The PTMs are regarded as a "knowledge base" (Petroni et al. 2019) to some extent.

However, it is possible that the PTMs do not fully grasp the tasks-specific information solely through pre-training. As a result, given the representation of the [MASK] tokens, we update the PTMs using contrastive learning by creating positive samples with instances from the same class and negative samples with instances from different classes, similar to the method in (Cui et al. 2022) as a plug-in module, yet without any additional encoder prior to the aforementioned re-embedding procedure, in order to provide the PTMs with task-specific knowledge as shown in Figure 3. Since a training instance can be represented by the embedding of the [MASK], denote $h_{ij},\ i \in 1, 2, ..., N; j \in 1, 2, ..., k$ as the set of training instances.

A positive instance pair is formed by an instance h_{ij} and another instance from the same class. A negative instance pair comprises two instances from different classes. The InfoNCE loss (Oord, Li, and Vinyals 2018) is adopted as the contrastive learning loss and the loss function is as follows.

$$\mathcal{L} = -\frac{1}{kN} \sum_{i=1}^{N} \sum_{j=1}^{k} \log \frac{\exp d(h_{ij}, h_{in})}{\sum_{p \neq i} \exp d(h_{ij}, h_{pq})} \tag{13}$$

where $d(\dots)$ is the distance metric in Formula 11.

Experiments

Datasets

We conduct experiments to demonstrate the effectiveness of our approach with 10 classification datasets (including 4 multi-class datasets) in both English and Chinese from GLUE (Wang et al. 2018), CLUE (Xu et al. 2020) and CBLUE (Zhang et al. 2022a) benchmarks. The datasets include 6 datasets from GLUE: SST-2 (Socher et al. 2013), MRPC (Dolan and Brockett 2005), QQP [2], QNLI (Rajpurkar et al. 2016) and RTE (Dagan, Glickman, and Magnini 2006), 3 datasets from CBLUE: CHIP-CTC (Zhang et al. 2022a), cMedTC (Zhang et al. 2020) and KUAKE-QIC (Zhang et al. 2022a) and 1 dataset from CLUE: Tnews (Xu et al. 2020). In particular, the datasets from CBLUE and CLUE are multi-class datasets. The statistics of the datasets are presented in Table 6 in Appendix . For the GLUE benchmark, we follow Gao et al. (Gao, Fisch, and Chen 2021) to use the original development sets as test sets and randomly select 16 instances for each class from the training set using 5 random seeds in few-shot scenarios and test with the full-size test set.

Baselines

We report our experimental results compared with the baselines with various verbalizer design methods, including manual verbalizer, automatic verbalizer and prototypical verbalizer. For a fair comparison, we fix the prompt template across various verbalizer methods and just investigate the performance with different verbalizer designs. We also experiment with the standard **Fine-tuning**. Manual Verbalizers (**ManualVerb**) are selected manually by domain experts empirically. Generated Token Verbalizers (**GenVerb**) are automatically searched from the vocabulary list of the PTMs. Here, we adopt the approach in LM-BFF (Gao, Fisch, and Chen 2021), which uses the conditional likelihood and re-ranking strategy to find the optimal token for the verbalizers. Prototypical Verbalizers (**ProtoVerb**) (Cui et al. 2022) are prototype embeddings directly learned for each class from the representation of training samples. During the inference process, the PTM makes the prediction by measuring the similarity between the query and every prototype embedding.

We implement the baselines with the Huggingface Transformers (Wolf et al. 2020) package based on PyTorch (Paszke et al. 2019) framework and OpenPrompt (Ding et al. 2022) toolkit. We adopt the RoBERTa-Large model (Liu et al. 2019; Cui et al. 2021) as our backbone PTM for both the English and Chinese tasks. Implementation details including prompt settings and experiment settings are in Appendix .

Parameter Updating

We introduced LLE-INC, a method that does not require additional parameters or parameter updating of the PTMs. In contrast, the baselines have different requirements. Vanilla Fine-tuning demands the training of a linear classifier, and GenVerb entails the exploration of optimal verbalizers and updating the PTMs with the verbalizers. ProtoVerb, although able to operate with frozen PTMs, necessitates the training of

[2] https://www.quora.com/q/quoradata/

	SST-2 (acc)	**MRPC** (F1)	**QQP** (F1)	**MNLI** (acc)	**QNLI** (acc)
ManualVerb†	54.6 (N/A)	63.5 (N/A)	63.2 (N/A)	32.3 (N/A)	49.5 (N/A)
LLE-INC†	86.6 (0.8)	67.6 (2.7)	66.1 (6.3)	58.1 (3.5)	62.4 (1.7)
Fine-tuning	81.4 (3.8)	76.6 (2.5)	60.7 (4.3)	45.8 (6.4)	60.2 (6.5)
ManualVerb	92.7 (0.9)	74.5 (5.3)	65.5 (5.3)	68.3 (2.3)	64.5 (4.2)
GenVerb	92.3 (1.0)	**76.2** (2.3)	67.0 (3.0)	68.3 (2.5)	68.3 (7.4)
ProtoVerb	87.8 (3.1)	66.9 (1.5)	66.9 (7.0)	62.1 (3.2)	59.9 (4.8)
LLE-INC	**92.9** (2.4)	**76.2** (3.1)	**68.1** (8.1)	**69.2** (1.9)	**70.2** (4.1)
	RTE (acc)	**CHIP-CTC** (acc)	**cMedTC** (acc)	**Kuake-QIC** (acc)	**Tnews** (acc)
ManualVerb†	47.7 (N/A)	9.4 (N/A)	10.4 (N/A)	2.3 (N/A)	6.1 (N/A)
LLE-INC†	68.2 (4.1)	44.5 (2.2)	47.7 (2.4)	31.2 (2.9)	41.3 (2.6)
Fine-tuning	54.4 (3.9)	29.0 (7.2)	25.2 (5.7)	13.1 (8.7)	13.5 (2.4)
ManualVerb	69.1 (3.6)	57.8 (1.8)	62.9 (3.2)	52.8 (3.4)	43.9 (2.2)
GenVerb	73.9 (2.2)	32.3 (1.9)	19.3 (14.2)	44.9 (2.9)	7.4 (4.6)
ProtoVerb	70.1 (3.7)	51.7 (2.4)	53.2 (7.7)	48.0 (0.9)	40.1 (1.5)
LLE-INC	**74.5** (1.9)	**61.0** (3.1)	**63.0** (1.2)	**54.8** (1.3)	**44.6** (1.1)

Table 1: Results with various prompt verbalizers. We report the mean (standard deviation) performance of accuracy/F1 over 5 random seeds. †: tuning-free. N/A: not applicable. Bold: best results.

	#p	#t-p	**SST-2**	**MRPC**	**QQP**	**MNLI**	**QNLI**	**RTE**
RoBERTa-LLE-INC	355M	355M	92.9	76.2	68.1	69.2	70.2	74.5
LLaMA-LLE-INC†	7B	0	93.2	81.4	77.3	66.5	73.2	74.5
LLaMA-w/o re-embed†	7B	0	88.0	79.3	70.3	57.0	71.2	72.2
LLaMA-LLE-INC†	13B	0	**93.8**	82.9	79.2	70.1	76.2	79.1
LLaMA-w/o re-embed†	13B	0	88.8	80.6	75.6	64.9	73.9	73.0
LLaMA-LLE-INC†	65B	0	92.8	**83.0**	**81.7**	**73.5**	**84.3**	**83.4**
LLaMA-w/o re-embed†	65B	0	86.8	81.2	77.4	66.2	79.1	73.6

Table 2: Experimental results for LLaMA-7B, LLaMA-13B and LLaMA-65B with LLE-INC. #p: the number of parameters. #t-p: the number of tuned parameters. †: tuning-free.

additional prototypical embeddings. Only ManualVerb is capable of directly operating on tasks following the pre-training process without tuning for downstream tasks. Consequently, we compare LLE-INC with the baselines under two conditions: one *with* parameter tuning and one *without* parameter tuning (including PTMs and additional parameters).

Results

We report the mean accuracy/F1 score following the baselines for each task across 5 sampled few-shot datasets using various random seeds, along with the standard deviation. The results for the experiments with and without parameter tuning are shown in Table 1.

Without Parameter Updating LLE-INC re-embeds the [MASK] representation without requiring the addition of any new parameters or parameter updates as introduced. In this situation, we only employ the frozen parameters in the PTM. Without any tuning parameters, only the ManualVerb in the baselines can still make predictions. The ManualVerb and LLE-INC which tune no parameter are denoted as ManualVerb† and LLE-INC†, respectively. Table 1 shows that ManualVerb† performs poorly without tuning, which means the model cannot understand the tasks and manually designed verbalizers explicitly. The performance of ManualVerb† on the test set is constant across different random seeds since the PTM is frozen. LLE-INC†, on the other hand, predicts the labels far more accurately than ManualVerb† and is on par with or even sometimes outperforms GenVerb and ProtoVerb (both demand parameter updating) because the space reconstruction of LLE-INC is based on the relationship with the intra-class neighbors. The aforementioned finding suggests that the output embeddings from the frozen PTM contain a significant amount of implicit information and manifold-based re-embedding is a viable approach to leverage the information.

With Parameter Updating Since the PTMs may not completely understand the task information, we can first train the models using contrastive learning. Then, we utilize the LLE-INC to re-embed the output representation of the updated PTMs as illustrated in Figure 3. Table 1 demonstrates that LLE-INC outperforms the baselines of Fine-tuning, GenVerb, ProtoVerb and even ManualVerb consistently and the

performance improvements are statistically significant with the p-value of paired t-test less than 0.05 in the majority of all cases. Our approach can bring up to 3.2% improvement on the CHIP-CTC dataset and 1.1% improvement on average across the 10 datasets. The experimental results of LLE-INC† reveal that the potential of PTMs can be reached by the manifold-based embedding space re-construction and the performance of LLE-INC further demonstrates that contrastive learning-based tuning for the parameters in the PTMs integrates better with the space re-embedding. As it comes to harder multi-class tasks, LLE-INC shows superiority to the baseline verbalizers without human effort for the re-embedding process can yield better representation for the instance. Meanwhile, we also notice that there can be duplicate verbalizers generated by GenVerb while dealing with multi-class datasets which negatively affects its performance.

Large Language Model with LLE-INC

The emergence of large language models (LLMs) has significantly transformed the field of natural language processing (NLP), leading to superior performance compared to earlier-generation paradigms. However, the fine-tuning of LLMs can pose a significant challenge due to the scale of their parameters. To address this challenge, we apply the LLE-INC to the tuning-free outputs of the LLaMA models (Touvron et al. 2023) (LLaMA-7B, LLaMA-13B, and LLaMA-65B) to explore the performance of LLE-INC. As LLaMA is primarily trained on English corpora, we only consider English language tasks. Our results, presented in Table 2, show that LLaMA-LLE-INC†, without any tuning parameters, can achieve better performance than RoBERTa-large with the PTM updating. This indicates that LLE-INC can be applied as a tuning-free method for classification tasks using large language models.

Analysis

Re-embedding Strategy LLE-INC re-embeds the verbalizer embedding space under the constraint of intra-class neighbors and makes predictions for the test instances with the k-nearest neighbors in the re-embedded space. Therefore, we also experiment with other re-embedding strategies: (1) LLE, the original locally linear embedding which only relies on the k-nearest-neighbor spatial relationship in the original space, and (2) w/o re-embedding, which makes predictions on the embeddings in the original space only with a kNN classifier. Experiments here do not include the contrastive learning module for a fair comparison. Table 3 shows that the performance of w/o re-embedding†is inferior to that of LLE† and LLE-INC†, indicating that the performance is limited by the original space. Furthermore, LLE† performs much better than w/o re-embedding† (up to 12.7% improvement) and LLE-INC† outperforms LLE† (up to 7.9% improvement) and demonstrates that the intra-class neighbor relationship is superior to spatial k-nearest neighbors.

Ablation Study An ablation study was conducted to evaluate the effectiveness of contrastive learning and space re-construction. The results in Table 4 demonstrate that the performance of LLE-INC in combination with contrastive learning surpasses that of either approach individually. This indicates that both parameter updating and embedding space reconstruction contribute to improved model performance.

	SST-2	MRPC	QQP
LLE-INC†	**86.6**	**67.6**	**66.1**
LLE†	79.2	61.4	62.1
w/o re-embed†	66.5	54.3	60.9
	MNLI	**QNLI**	**RTE**
LLE-INC†	**58.1**	**62.4**	**68.2**
LLE†	51.9	58.0	60.3
w/o re-embed†	42.4	54.7	61.1
	CHIP-CTC	**Kuake-QIC**	**Tnews**
LLE-INC†	**44.5**	**31.2**	**41.3**
LLE†	39.2	29.6	39.5
w/o re-embed†	33.8	28.1	40.3

Table 3: Experimental results for different re-embedding strategies. †: tuning-free.

	SST-2	MRPC	QQP
LLE-INC with CL	**92.9**	**76.2**	**68.1**
LLE-INC	86.6	67.6	66.1
CL	85.1	64.2	66.5
	MNLI	**QNLI**	**RTE**
LLE-INC with CL	**69.2**	**70.2**	**74.5**
LLE-INC	58.1	62.4	68.2
CL	57.0	60.4	67.8
	CHIP-CTC	**Kuake-QIC**	**Tnews**
LLE-INC with CL	**61.0**	**54.8**	**44.6**
LLE-INC	44.5	31.2	41.3
CL	49.7	40.1	38.2

Table 4: Ablation study for the LLE-INC and contrastive learning. CL: contrastive learning.

Conclusion

Prompt-based tuning has been proven effective in few-shot scenarios and recently embedded verbalizers have been explored as an alternative to the labor-intensive process of existing verbalizers. Recent studies are dependent on the tuning of the PTM or extra trainable embeddings and the manifold in the high-dimensional representation space has the potential to mislead Euclidean distance measurements. In this study, we propose to re-embed the verbalizer representation space through locally linear embedding with an intra-class neighborhood constraint. Experimental results demonstrate that LLE-INC works rather well without any parameter tuning and can further enhance the performance of prompt-based tuning in conjunction with model tuning. The effectiveness of manifold learning on the LLaMA model also opens up new possibilities for the tuning-free application of LLMs and further inspires computational resource-friendly research on the LLMs.

Acknowledgements

We thank the anonymous reviewers for their insightful and constructive comments and gratefully acknowledge the support of the National Key R&D Program of China (2021ZD0113302), the National Natural Science Foundation of China Youth Fund (62206079), and the Heilongjiang Provincial Natural Science Foundation of China (YQ2022F006). We also appreciate the support from Du Xiaoman (Beijing) Science Technology Co., Ltd. on our research.

References

Cui, G.; Hu, S.; Ding, N.; Huang, L.; and Liu, Z. 2022. Prototypical Verbalizer for Prompt-based Few-shot Tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7014–7024.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; and Yang, Z. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 3504–3514.

Dagan, I.; Glickman, O.; and Magnini, B. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, 177–190. Springer.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Ding, N.; Hu, S.; Zhao, W.; Chen, Y.; Liu, Z.; Zheng, H.; and Sun, M. 2022. OpenPrompt: An Open-source Framework for Prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 105–113.

Dolan, B.; and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Du, Y.; Zhao, S.; Wang, H.; Chen, Y.; Bai, R.; Qiang, Z.; Cai, M.; and Qin, B. 2023. Make Your Decision Convincing! A Unified Two-Stage Framework: Self-Attribution and Decision-Making. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 1101–1112.

Gao, T.; Fisch, A.; and Chen, D. 2021. 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)*, 3816–3830. Online: Association for Computational Linguistics.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6894–6910.

Hambardzumyan, K.; Khachatrian, H.; and May, J. 2021. WARP: Word-level Adversarial ReProgramming. 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)*, 4921–4933.

Hasan, S.; and Curry, E. 2017. Word re-embedding via manifold dimensionality retention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 321–326.

Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339. Melbourne, Australia: Association for Computational Linguistics.

Hu, S.; Ding, N.; Wang, H.; Liu, Z.; Wang, J.; Li, J.; Wu, W.; and Sun, M. 2022. Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2225–2240.

Jiang, T.; Huang, S.; Zhang, Z.; Wang, D.; Zhuang, F.; Wei, F.; Huang, H.; Zhang, L.; and Zhang, Q. 2022. PromptBERT: Improving BERT Sentence Embeddings with Prompts. *arXiv preprint arXiv:2201.04337*.

Lee, J. 2010. *Introduction to topological manifolds*, volume 202. Springer Science & Business Media.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. 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)*, 4582–4597. Online: Association for Computational Linguistics.

Liu, C.; Wang, H.; Xi, N.; Zhao, S.; and Qin, B. 2023. Global Prompt Cell: A Portable Control Module forÂ Effective Prompt Tuning. In *Natural Language Processing and Chinese Computing*, 657–668.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021b. GPT understands, too. *arXiv preprint arXiv:2103.10385*.

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

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 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)*, 2463–2473.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392.

Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326.

Saul, L. K.; and Roweis, S. T. 2000. An introduction to locally linear embedding. *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html*.

Schick, T.; and Schütze, H. 2021a. 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*, 255–269.

Schick, T.; and Schütze, H. 2021b. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2339–2352.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. Eliciting Knowledge from Language Models Using Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4222–4235.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *International Conference on Learning Representations*.

Wang, B.; Sun, Y.; Chu, Y.; Lin, H.; Zhao, D.; Yang, L.; Shen, C.; Yang, Z.; and Wang, J. 2022a. Manifold biomedical text sentence embedding. *Neurocomputing*, 492: 117–125.

Wang, H.; Liu, C.; Xi, N.; Zhao, S.; Ju, M.; Zhang, S.; Zhang, Z.; Zheng, Y.; Qin, B.; and Liu, T. 2022b. Prompt Combines Paraphrase: Teaching Pre-trained Models to Understand Rare Biomedical Words. In *Proceedings of the 29th International Conference on Computational Linguistics*, 1422–1431.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Xu, L.; Hu, H.; Zhang, X.; Li, L.; Cao, C.; Li, Y.; Xu, Y.; Sun, K.; Yu, D.; Yu, C.; et al. 2020. CLUE: A Chinese Language Understanding Evaluation Benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4762–4772.

Yonghe, C.; Lin, H.; Yang, L.; Diao, Y.; Zhang, S.; and Xiaochao, F. 2019. Refining Word Representations by Manifold Learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5394–5400. International Joint Conferences on Artificial Intelligence Organization.

Yu, Y.; Zuo, S.; Jiang, H.; Ren, W.; Zhao, T.; and Zhang, C. 2021. Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1063–1077.

Zhang, N.; Chen, M.; Bi, Z.; Liang, X.; Li, L.; Shang, X.; Yin, K.; Tan, C.; Xu, J.; Huang, F.; et al. 2022a. CBLUE: A Chinese Biomedical Language Understanding Evaluation Benchmark. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7888–7915.

Zhang, N.; Jia, Q.; Yin, K.; Dong, L.; Gao, F.; and Hua, N. 2020. Conceptualized representation learning for chinese biomedical text mining. *arXiv preprint arXiv:2008.10813*.

Zhang, N.; Li, L.; Chen, X.; Deng, S.; Bi, Z.; Tan, C.; Huang, F.; and Chen, H. 2022b. Differentiable Prompt Makes Pre-trained Language Models Better Few-shot Learners. In *International Conference on Learning Representations*.