Feature Normalization and Cartography-Based Demonstrations for Prompt-Based Fine-Tuning on Emotion-Related Tasks

Mahshid Hosseini, Cornelia Caragea

Computer Science
University of Illinois Chicago
mhosse4@uic.edu, cornelia@uic.edu

Abstract

To train a model in a traditional supervised learning classification system for natural language processing (NLP) tasks, it is essential to have labeled data, which is not present in large amounts for many tasks. Prompt-based learning methods attempt to combat the supervised learning need for labeled data by directly adapting pre-trained language models and modeling the probability of text itself. In this paper, we propose a novel data-agnostic strategy for prompt-based fine-tuning that leverages feature moments (a.k.a., mean and standard deviation) as a data augmentation technique and employs training dynamics (i.e., confidence and variability) to allow more informative samples to be concatenated for generating demonstrations as input context. Our approach is a strong method for few-shot learning that forces the language model to pay special attention to the feature moments and allows more informative samples to be concatenated for generating demonstrations as input context by selecting high confidence and low variance samples. To demonstrate its effectiveness given limited training data, we conduct extensive experiments in different few-shot settings on three empathy and emotion classification datasets (from various domains). We further evaluate our method's robustness by introducing noise to our few-shot input data and labels and show that exchanging moments between samples and incorporating cartography-based demonstrations are beneficial when the available data is limited and noisy.

Introduction

Despite the fact that pre-training on a large corpus of text has manifested substantial gains on many NLP tasks (Liu et al. 2019a; Devlin et al. 2019; Radford et al. 2018), effectively employing such paradigms depends on the presence of tens of thousands of labeled samples. However, the time-consuming and labor-intensive annotation process has made it challenging to obtain large labeled datasets in real-world scenarios like computer-assisted therapy sessions where the relevant empathy or emotion data might not be present at a large scale (Hosseini and Caragea 2021a). In recent years, a resurgence of work in few-shot learning has led to major advances in natural language understanding (NLU) tasks. Strikingly, GPT-3 model (Brown et al. 2020) has shown remarkable few-shot capabilities only by employing a natural-language prompt and some task demonstrations as input context. However, GPT-3 is a humongous autoregressive language model with 175 billion parameters, which makes it inexpedient to use in most real-world applications.

Following the route of prompt-based prediction, Gao, Fisch, and Chen (2021) studied few-shot learning in a more realistic scenario, where they employed a moderately-sized language model like RoBERTa (Liu et al. 2019b) or BERT (Devlin et al. 2019) for which fine-tuning is computationally efficient. In addition, Gao, Fisch, and Chen (2021) explored fine-tuning with demonstrations where they randomly select a single sample from each class that is semantically close to the input sample to form the demonstrations.

Inspired by these observations, we propose a novel strategy for prompt-based fine-tuning that (1) leverages feature moments (a.k.a., mean and standard deviation) as a data augmentation technique and (2) generates demonstrations based on the characteristics of the data samples and their contextual similarity to the input sample. Our approach is a strong data-agnostic method for few-shot learning that forces the language model to pay special attention to the feature moments and allows more informative samples to be concatenated for generating demonstrations as input context by selecting high confidence and low variance samples. To leverage feature moments (Li et al. 2021), our approach pulls the mean and variance (across feature dimensions) of the hidden state representations and swaps them between samples. In other words, we extract and detach the feature moments of an input sample x_i and subsume the moments of another sample x_j (in the same mini-batch) under x_i. Consequently, resulting features emanating from the moment exchange process contain information about both samples, which in turn makes the few-shot model more robust by paying attention to two salient signals, i.e., the normalized feature of x_i and the moments of x_j.

Furthermore, our approach leverages training dynamics (i.e., the model's behavior as training progresses) to generate informative demonstrations targeted at improving prompt-based fine-tuning with a small number of annotated examples. To this end, our approach first employs data maps (Swayamdipta et al. 2020) to characterize training samples to different groups, namely easy-to-learn, ambiguous, and hard-to-learn. Then, it selects samples based on how they contribute to the model learning and their semantic similarity to the input sample. Our goal, by employing such particularities, is to impart the model with the samples with high

confidence and low variability as demonstrations, which are the most informative data samples. To demonstrate the effectiveness of our proposed approach given limited training data, we conduct extensive experiments on three emotion and empathy classification datasets (where annotated data is often difficult to obtain for in real-world scenarios) and examine the performance of the pre-trained language model, RoBERTa (Liu et al. 2019b), in different few-shot settings. We further evaluate our method's robustness by introducing noise to our few-shot input data and labels and show that exchanging moments between samples and incorporating cartography-based demonstrations are beneficial when the available data is limited and noisy. In order to introduce noise to the input samples, we use a specific data augmentation (Wei and Zou 2019) technique which, for a given sentence in the training set, randomly deletes, replaces, swaps, or inserts a random word in the sentence. We also leverage training dynamics (Swayamdipta et al. 2020) and extract the samples that are detected as weakly labeled or mislabeled (i.e., hard-to-learn instances) and use them as our few-shot samples in the training set as label errors.

Our contributions are three-folded: (1) We propose a novel data-agnostic approach for fine-tuning language models with a small number of annotated examples on the emotion-related tasks. Our approach leverages moment exchange as a data augmentation technique to make the network utilize the salient signals in the feature moments to improve prompt-based fine-tuning accuracy and robustness. To the best of our knowledge, we are the first that utilize feature moments for prompting for fine-tuning on downstream tasks; (2) We further propose a simple yet effective cartography-based strategy that leverages training dynamics (confidence and variability) to allow more informative samples to be concatenated for generating demonstrations as input context; (3) Through extensive experiments in different few-shot settings with noisy data, we empirically show that our approach can consistently improve prompt-based fine-tuning accuracy and robustness across three emotion and empathy classification datasets.

Related Work

A large and growing body of research has been conducted on training very large neural networks for language understanding across various tasks (Liu et al. 2019a; Devlin et al. 2019; Radford et al. 2018; Raffel et al. 2020). Although these models achieved remarkable results across a wide range of natural language tasks, effectively employing such paradigms depends on the presence of tens of thousands of task-specific training examples to fine-tune the model. GPT-3 model (Brown et al. 2020), on the other hand, has shown remarkable capabilities for few-shot predictions only by employing a natural language task description and some task demonstrations as input context. However, as an extremely large autoregressive language model, it is challenging to use GPT-3 in most real-world applications.

Following GPT-3 in-context learning method where *many* demonstrations (depending on the model's context window size) are randomly selected from the training set, Gao, Fisch, and Chen (2021) proposed to append *only one* selected training sample from each class to the input based on their similarity to the input sample. In this way, the final input sample becomes shorter and easier to be leveraged by a medium-sized language model like RoBERTa. We are greatly inspired by (Gao, Fisch, and Chen 2021), although they *randomly* pick a single example from each class to create demonstrations and select to pair inputs with similar examples. We, on the other hand, *leverage training dynamics* to generate informative demonstrations based on the characteristics of the data samples and their contextual similarity to the input sample. We also *leverage feature moments* (i.e., mean and standard deviation) as a data augmentation technique and provide the language model with two salient signals, the normalized features and the moments of samples.

Along the lines of prompt-based learning, there are several studies on prompting for extracting knowledge from pre-trained models (Petroni et al. 2019; Davison, Feldman, and Rush 2019; Talmor et al. 2020). Petroni et al. (2019) proposed a probe for examining the commonsense knowledge contained in pre-trained language models. Petroni et al. (2019) showed that BERT-large (Devlin et al. 2019) can effectively evoke such knowledge in comparison with non-neural and supervised alternatives. Davison, Feldman, and Rush (2019) also proposed an unsupervised technique for commonsense knowledge base completion by employing the pre-trained language models' knowledge. Talmor et al. (2020) introduced various tasks for examining the reasoning capabilities of different language models. Schick and Schütze (2021) proposed a semi-supervised learning method that transforms input examples to cloze-style phrases and leveraged the generated phrases to predict soft labels for a set of unlabeled samples. Finally, Schick and Schütze (2021) performed supervised training on the resulting training set. As an extension to the previous work, Schick, Schmid, and Schütze (2020) further introduced a technique that automatically maps labels to words. Jiang et al. (2020) also introduced paraphrasing and mining-based approaches to automatically generate prompts with good quality and diversity. Evaluating on eight different tasks, Radford et al. (2019) showed that large language models have capacity to perform well on down-stream tasks in zero-shot setting. Zhu et al. (2021) proposed a unified perception architecture for the zero-shot inference that leverages different modalities. Zhou et al. (2022) studied parameter-efficient prompt learning, analyzed the generalizability issue of static prompts, and showed that using conditional prompt learning to generate a simple design can perform very well in various problem scenarios.

Data

We perform evaluations on three text classification datasets on emotion detection and empathy classification. Specifically, we use the emotion detection datasets by Poria et al. (2019) and Bostan, Kim, and Klinger (2020) and the empathy direction detection dataset by Hosseini and Caragea (2021b). We select to perform evaluations on these datasets since they are annotated at the sentence level and fit prompting for fine-tuning on a downstream task better. Other emotion-related datasets with multiple sentences (e.g., paragraphs) are usually composed of multiple topics, not necessarily conveying emotion or empathy in all the sentences, and often switching

between these topics from one sentence to another, which is not suitable for the prompt-based fine-tuning scenario. Next, we describe the datasets:

- MELD: MELD (Poria et al. 2019) includes dialogues from TV-series Friends, annotated with Ekman's six emotions (Ekman 1992) and two additional emotion labels of neutral and non-neutral. We use Ekman-6 emotions and neutral as these were publicly available.
- GoodNewsEveryone: Bostan, Kim, and Klinger (2020) collected a dataset of English news headlines gathered from different news sources. This dataset is annotated with anger, annoyance, disgust, fear, guilt, joy, love, surprise, anticipation, pride, sadness, shame, and trust.
- iEmpathize: Hosseini and Caragea (2021b) collected an empathy direction detection dataset from an online cancer forum which is annotated with three categories of seeking-empathy (seek), providing-empathy (provide), or none.

To evaluate our method's robustness against noise, we conduct the evaluations on three different settings (detailed below) and build our few-shot training set by randomly selecting $\mathbb{K}$ samples per class for each setting.

MAIN: In this setting, the main original samples from the dataset are used in training set.

NOISYS: Real-world data often contain noise. In this setting, instead of using the main samples from the dataset, we augment the training set using the easy data augmentation techniques (Wei and Zou 2019) intending to introduce noise to the input samples. EDA consists of four operations: random deletion, synonym replacement, random swap, and random insertion. For a given sentence in the training set, EDA randomly performs at least one of these operations. For instance, after applying EDA, *"My father has lost* 32 *lbs since mid April and is down to* 118*."* is replaced by *"my has lost lbs since mid april and is down"*.

NOISYL: The contribution of the training samples to the generalization is not equal. In other words, some of the samples may be more representative of the label class, and some can be ambiguous. In this setting, we extract the samples that are detected to be weakly labeled or mislabeled by the model and use them as part of our few-shot samples in the training set. In particular, we leverage training dynamics as described in the Data Characterization Section to detect samples with low variability and low confidence that correspond to hard-to-learn examples. Swayamdipta et al. (2020) showed that such instances are seldom predicted correctly during training and often correspond to data errors. The following samples are samples within the hard-to-learn category (detected to be mislabeled) from the empathy dataset: *Does the cardiologist think any permanent damage has been done?* (none label), *When I left the hospital the first time they gave 13% chance to live.* (seeking-empathy label), and *Plan as though you had mere weeks left.* (providing-empathy label).

Task Setup

Problem formulation. Given a pre-trained language model $\mathbb{L}$, our goal is to fine-tune $\mathbb{L}$ on the $\mathbb{Y}$-class classification task $\mathbb{D}$ (i.e., emotion or empathy classification). In the few-shot setting, we assume access to only $\mathbb{K}$ training examples per class which constitutes our training set $\mathbb{D}_{train}$[1]. The goal is to generalize well on an unseen test set $\mathbb{D}_{test}$. To guarantee that we are learning from limited data, we also employ a small development set $\mathbb{D}_{dev}$ that also contains $\mathbb{K}$ samples per class similar to our few-shot training set. For the experiments, we use $\mathbb{L}$ = RoBERTa-base and experiment with $\mathbb{K} = [4, 8, 16]$.

Prompt-based Fine-tuning

Here we use the empathy detection task to describe the fine-tuning method. The same would be applied to the emotion detection task with the proper prompt. Let us consider *"I'm never going to give up, please continue to pray for him."* as our input sample x_1, by prompting a language model $\mathbb{L}$, we can formulate the empathy detection task as:

x_{prompt} = `[CLS]` x_1 I am so `[MASK]`. `[SEP]`

and ask $\mathbb{L}$ to fill the `[MASK]` with a proper word, i.e., *"sad"* (seek), *"sorry"* (provide), *"great"* (none). Therefore, for each input sample x_{in}, we generate a manipulated input x_{prompt} which contains one `[MASK]` token as our masked language modeling input. In this fashion, we can serve our empathy classification task as masked language modeling which predicts the probability $y \in \mathbb{Y}$ as:

$$p(y|x_{in}) = p(\texttt{[MASK]} = \mathbb{M}(y)|x_{prompt})$$

where $\mathbb{M} : \mathbb{Y} \rightarrow \mathbb{V}$ provides the mapping between label space $\mathbb{Y}$ and words in the vocabulary $\mathbb{V}$ of the language model $\mathbb{L}$.

Automatic Prompt Generation. Manually designing optimal prompts (i.e., templates and label words) is a challenging and time-consuming task that usually demands domain knowledge and interpreting of the language model's internal processes. To address this complication and generate templates automatically, we use the pre-trained text-to-text transfer Transformer T5 (Raffel et al. 2020) model. T5 randomly selects 15% of the tokens in the input sequence to drop out, replaces them with mask tokens (e.g., $\langle \texttt{X} \rangle$, $\langle \texttt{Y} \rangle$) and is pre-trained to predict the dropped-out tokens. Using the input samples from $\mathbb{D}_{train}$, we can employ this procedure for prompt generation and have T5 to generate the template $\mathbb{T}$. Following Gao, Fisch, and Chen (2021), we use the below conversions for our training samples to form the T5 inputs ($\langle S \rangle$ refers to the input sample):

$$\tilde{\mathbb{T}}(x_{in}, y) = \begin{cases} \langle S \rangle \rightarrow \langle \texttt{X} \rangle \mathbb{M}(y) \langle \texttt{Y} \rangle \langle S \rangle, \\ \langle S \rangle \rightarrow \langle S \rangle \langle \texttt{X} \rangle \mathbb{M}(y) \langle \texttt{Y} \rangle \end{cases}$$

T5 predicts missed tokens aiming to maximize:

$$\sum_{(x_{in}, y) \in \mathbb{D}_{train}} log \mathbb{P}_{T5}(\mathbb{T}|\tilde{\mathbb{T}}(x_{in}, y))$$

where $\mathbb{P}_{T5}$ is the T5 output probability distribution. In this way, we aim to find a template $\mathbb{T}$ that fits all training samples perfectly. Then, we get various templates using beam search (with beam width 100), fine-tune the generated templates on our training samples, and select the template with the best performance on $\mathbb{D}_{dev}$. Table 1 present the manual and top 3 automatically generated templates for MELD dataset for each setting (i.e., Main, NoisyS, and NoisyL) for $\mathbb{K} = [4, 8, 16]$. The first automatically generated template for each setting has the highest performance and is used in our experiments.

[1]The total size of the training set is $|\mathbb{D}_{train}| = \mathbb{K} \times |\mathbb{Y}|$, where $|\mathbb{Y}|$ represents the number of classes in $\mathbb{Y}$.

MELD	anger/disgust/ fear/joy/ sadness/ surprise/ neutral		
Manual $\mathbb{T}$	It was `[MASK]`. $\langle S \rangle$.		
Automatic $\mathbb{T}$	$\mathbb{M}(\mathbb{Y})$ = {angry, disgusting, frightening, joyful, sad, surprising, neutral}		
	MAIN	NOISYS	NOISYL
$\mathbb{K} = 4$	1. $\langle S \rangle$ It's `[MASK]`!	1. $\langle S \rangle$ Too `[MASK]`!	1. $\langle S \rangle$ It's `[MASK]`!
	2. How `[MASK]`. $\langle S \rangle$.	2. Just `[MASK]`! $\langle S \rangle$.	2. $\langle S \rangle$ I'm `[MASK]`.
	3. $\langle S \rangle$ Are you `[MASK]`?	3. $\langle S \rangle$ Is it `[MASK]`?	3. $\langle S \rangle$ That is `[MASK]`.
$\mathbb{K} = 8$	1. $\langle S \rangle$ So `[MASK]`!	1. That's `[MASK]`. $\langle S \rangle$	1. Really `[MASK]`! $\langle S \rangle$.
	2. $\langle S \rangle$ It was `[MASK]`.	2. Very `[MASK]`! $\langle S \rangle$.	2. A little `[MASK]`! $\langle S \rangle$.
	3. $\langle S \rangle$ That was `[MASK]`.	3. $\langle S \rangle$ Sounds `[MASK]`!	3. $\langle S \rangle$ It was `[MASK]`!
$\mathbb{K} = 16$	1. $\langle S \rangle$ Really `[MASK]`!	1. Very `[MASK]`. $\langle S \rangle$.	1. Really `[MASK]`. $\langle S \rangle$.
	2. $\langle S \rangle$ It was `[MASK]`.	2. $\langle S \rangle$ It was `[MASK]`.	2. $\langle S \rangle$ This is `[MASK]`.
	3. $\langle S \rangle$ It is `[MASK]`.	3. $\langle S \rangle$ Too `[MASK]`.	3. $\langle S \rangle$ It is `[MASK]`!

Table 1: Top 3 automatically generated templates for each $\mathbb{K}$ value for MELD dataset. Manual template is kept the same for all the settings.

Moment Exchange Data Augmentation for Prompt-based Fine-tuning

Here we propose to use the moments (a.k.a., mean and standard deviation) of latent features as an implicit data augmentation technique to improve our prompt-based fine-tuning in the few-shot setting. Leveraging moments has been initially introduced for computer vision tasks by Li et al. (2021) to enhance the performance of image classification and generation models. In essence, moment exchange (i.e., MoEx) (Li et al. 2021) combines feature normalization with data augmentation and views moments as features (not noise like in the case of batch normalization (Ioffe and Szegedy 2015) in the computer vision tasks). Inspired by the moment exchange for image classification, in this paper, we propose to leverage such statistics (i.e., mean and standard deviation) of the textual samples for prompt-based fine-tuning.

In transformers, layer normalization computes the mean μ and standard deviation (std) σ across all features and all elements (i.e., words) for each instance (i.e., sentence) independently and normalizes the features with these statistics and a stability constant ϵ.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{and} \quad \sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2} \tag{1}$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{2}$$

Inspired by Li et al. (2021), we pull the mean and variance (across the feature dimension) of the hidden feature representations and exchange them between samples. In other words, we extract and detach the feature moments of input sample x_i and subsume the moments of another sample x_j (in the same mini-batch) under x_i. Having μ_i, σ_i, μ_j, σ_j as instance-dependent moments (retaining label-relevant signals and representing the underlying structure of the samples), we insert the sample x_j moments into the sample x_i feature representation as $h_i^{(j)} = \sigma_j \frac{h_i - \mu_i}{\sigma_i} + \mu_j$ (as depicted in the Figure 1 upper part) and change the loss function to predict the labels y_i and y_j, based on a mixing parameter $\lambda \in [0, 1]$:

$$\lambda \mathcal{L}(h_i^{(j)}, y_i) + (1 - \lambda)\mathcal{L}(h_i^{(j)}, y_j) \tag{3}$$

In such manner, we confer the resulting features to contain information about both samples and push the model to predict an interpolation of the two labels. Consequently, we impart the model with two salient signals, i.e., the normalized feature of x_i and the moments of x_j, which in turn help to increase the robustness of the prompt-based fine-tuning.

CBDemo: Fine-tuning with Cartography-Based Demonstrations

We introduce our proposed CBDemo, a simple yet effective method to leverage demonstrations targeted at improving prompt-based fine-tuning that generates demonstrations based on training dynamics (Swayamdipta et al. 2020).

Our proposed CBDemo first leverages training dynamics to characterize each training sample based on how they contribute to the model learning and then sample examples with specific characteristics (emanated from the previous step) that are semantically close to x_{in}. For calculating training dynamics statistics, we use RoBERTa-base (Liu et al. 2019b) with the same set of hyper-parameters as (Swayamdipta et al. 2020). Below we describe our fine-tuning strategy with cartography-based demonstrations.

Data Characterization

We first compute training dynamics statistics, i.e., confidence and variability (Swayamdipta et al. 2020), to stratify the training set $\mathcal{D}_{train}$ into three different categories, namely easy-to-learn, ambiguous, and hard-to-learn. These statistics are calculated across the E training epochs for each instance i

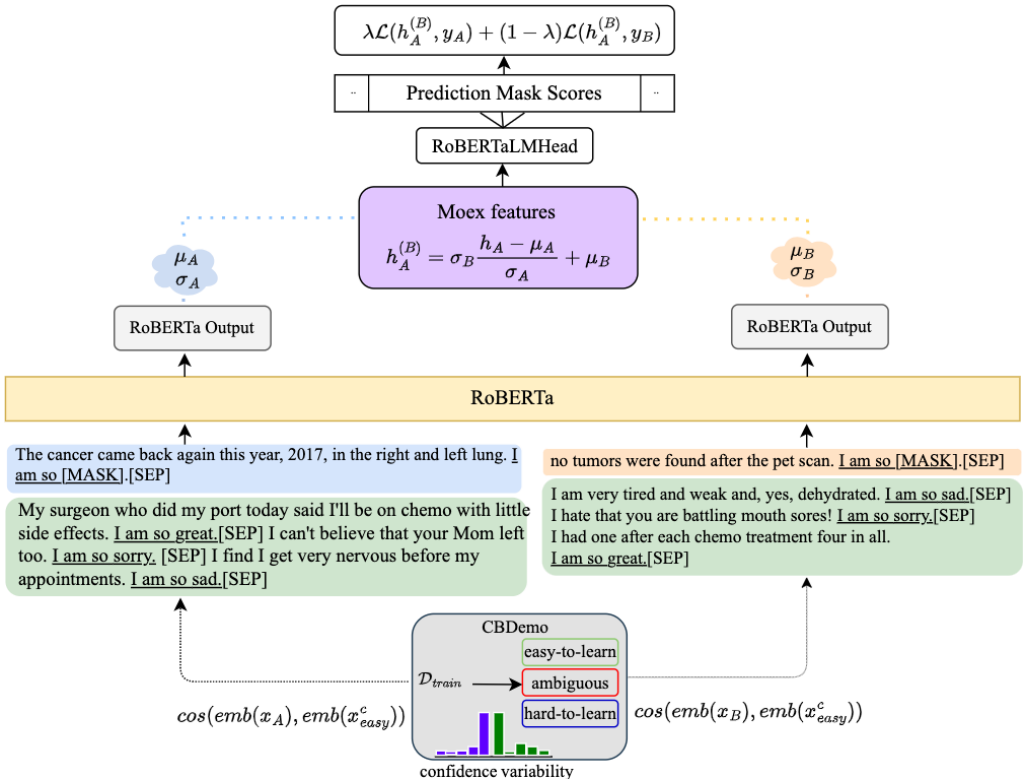

Figure 1: Our proposed method for prompt-based fine-tuning with moment exchange and cartography-based demonstrations. The samples in the green box are automatically selected by CBDemo from the easy-to-learn category based on their contextual similarity to the query samples in the blue or orange box.

and its true label, (x_i, y_i). Confidence is defined as the mean model probability of the true label y_i across epochs:

$$\hat{\mu}_i = \frac{1}{E}\sum_{e=1}^{E} p_{\theta_e}(y_i|x_i) \tag{4}$$

where p_{θ_e} indicates the model's probability at the end of the e_{th} epoch with θ being our model parameters.

Variability is calculated as the standard deviation of the ground-truth probabilities $p_{\theta_e}(y_i|\hat{x}_i)$ across different epochs:

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{e=1}^{E}(p_{\theta_e}(y_i|x_i) - \hat{\mu}_i)}{E}} \tag{5}$$

Using these statistics, which are derived from the model's behavior on individual instances across training, we select the easy-to-learn samples (i.e., samples that the model *consistently* predicts *correctly* across epochs) to generate our demonstrations. Our goal, by employing such particularities, is to impart the model with the samples with high confidence and low variability as demonstrations, which are the most informative data samples.

To control the construction of the demonstration examples (x^c_{easy}, y^c) (c refers to class), and assure that the group of selected demonstrations are not significantly divergent from the input x_{in} or from each other, we sample easy-to-learn examples that are semantically close to x_{in}. To this end, we first derive the embeddings of all input sentences using a pre-trained SBERT (Reimers and Gurevych 2019) and calculate the similarity score of the input example and the training instances of a particular class belonging to the easy-to-learn category, $cos(emb(x_{in}), emb(x^c_{easy}))$. We empirically select to use top 60% most similar examples for each class to use as demonstrations.

At each training step, CBDemo selects the most similar easy-to-learn example (x^c_{easy}, y^c) from each class and generates a template $\hat{\mathbb{T}}(x^c_{easy}, y^c)$, by converting it into $\mathbb{T}(x^c_{easy})$ with `[MASK]` substituted by $\mathbb{M}(y^c)$. Then, it concatenates the generated templates with input sample x_{in}:

$$\mathbb{T}(x_{in}) +\!\!+ \hat{\mathbb{T}}(x^1_{easy}, y^1) +\!\!+ \cdots +\!\!+ \hat{\mathbb{T}}(x^{|\mathbb{Y}|}_{easy}, y^{|\mathbb{Y}|})$$

where $+\!\!+$ indicates concatenation of input sequences.

Experiments

Baseline Methods

The details of the experiments are as follows. In all the experiments, we use RoBERTa-base and $\mathbb{K} = [4, 8, 16]$. We contrast our proposed approach that uses manual and automatically generated templates by T5 model, with the baselines: (1) prompt-based zero-shot prediction using the manual prompts without any training samples; (2) GPT-3 in-context learning that randomly selects 32 demonstrations and adds them to the input with the prompt-based zero-shot setting; (3) few-shot standard fine-tuning; (4) PET (Schick and Schütze 2021) that is a semi-supervised training approach for prompt-based fine-tuning; (5) prompt-based fine-tuning without demonstrations;

Model	MELD								
	K = 4			K = 8			K = 16		
	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL
Prompt-based zero-shot**	16.54**								
GPT-3 in-context learning	12.32	10.58	12.60	14.93	12.98	13.16	14.50	13.61	11.00
Fine-tuning (few-shot)	28.95	27.77	12.74	31.10	25.06	13.94	34.80	31.92	15.49
PET	28.10	24.84	10.09	29.45	28.45	11.26	31.28	31.23	10.12
Prompt-based FT (manual)	30.35	30.17	13.12	30.65	28.98	13.88	35.70	33.45	11.63
+ random_demo	40.93	23.99	13.05	40.57	25.79	12.58	44.17	32.83	13.68
+ sim_demo	43.37	26.30	12.37	38.14	24.15	13.11	40.93	31.31	15.21
+ ours	48.68	37.10	18.80	52.37	39.92	29.91	47.72	42.26	18.85
Prompt-based FT (automatic)	32.88	26.33	15.50	31.37	30.65	13.52	39.18	36.88	11.09
+ random_demo	38.94	27.20	17.84	37.60	27.84	14.15	46.97	30.58	14.33
+ sim_demo	36.60	26.24	12.48	39.94	25.48	13.19	47.42	31.68	15.45
+ ours	43.70	41.90	19.90	47.80	38.95	24.05	50.25	41.05	20.32
Fine-tuning (full)**	62.06**								

Table 2: Accuracy on emotion dataset (i.e., MELD) using RoBERTa-base. In all the few-shot settings, we use $\mathbb{K} = [4, 8, 16]$ samples per class; Fine-tuning (full)** uses the full training set and is the same for all our three settings; Prompt-based zero-shot** uses no training samples and is same for all our three settings; manual and automatic refer to our manual and automatically generated templates, respectively; random_demo and sim_demo (Gao, Fisch, and Chen 2021) refer to random and similarity-based demonstrations, respectively; NOISYS and NOISYL refer to noisy samples and noisy labels, respectively.

Model	GoodNewsEveryone								
	K = 4			K = 8			K = 16		
	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL
Prompt-based zero-shot**	21.42**								
GPT-3 in-context learning	16.80	16.11	14.88	16.52	15.60	13.08	16.90	16.22	13.54
Fine-tuning (few-shot)	15.66	15.54	12.32	17.76	16.42	16.26	20.98	20.50	20.42
PET	20.57	21.09	17.46	22.40	24.12	21.58	23.27	22.39	20.14
Prompt-based FT (manual)	22.90	22.30	20.80	22.96	26.03	22.48	27.18	25.40	26.18
+ random_demo	23.46	15.12	12.44	24.68	16.34	17.58	29.94	20.26	20.28
+ sim_demo	22.92	15.32	13.00	24.64	15.40	16.12	30.88	19.22	21.66
+ ours	25.75	28.98	24.45	31.50	30.12	29.02	34.87	32.12	29.98
Prompt-based FT (automatic)	23.94	21.38	20.06	23.86	26.08	23.78	26.28	26.16	26.70
+ random_demo	26.10	15.92	12.45	25.74	16.76	16.62	30.42	20.08	19.98
+ sim_demo	26.78	14.72	13.20	26.76	15.52	15.50	32.24	21.20	20.74
+ ours	31.50	27.05	24.58	30.10	28.75	26.90	35.69	33.21	31.46
Fine-tuning (full)**	36.14**								

Table 3: Accuracy on emotion dataset (i.e., GoodNewsEveryone) using RoBERTa-base. For the definitions refer to the Table 2.

(6) prompt-based fine-tuning with random demonstrations; (7) prompt-based fine-tuning with similarity-based demonstrations (Gao, Fisch, and Chen 2021); and (8) standard fine-tuning with the full training set.

Results

Tables 2, 3, and 4 present the comparison of our proposed approach and baseline methods on emotion (i.e., MELD, GoodNewsEveryone) and empathy (i.e., iEmpathize) classification tasks, respectively. We report the average performance on 5 distinct randomly sampled training and dev splits with five random seeds to provide a robust measure of our few-shot performance. We make a few remarks below.

As we can see from the tables, our proposed method achieves higher accuracy on all the few-shot settings than any baseline using both the designed and generated prompts. From Tables 2, 3, and 4, we can observe that exchanging moments between samples through MoEx and incorporat-

Model	iEmpathize								
	K = 4			K = 8			K = 16		
	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL
Prompt-based zero-shot**	25.40**								
GPT-3 in-context learning	22.36	20.18	20.01	22.95	22.03	21.55	24.00	22.12	21.40
Fine-tuning (few-shot)	56.40	55.20	22.00	60.20	59.60	25.20	67.30	63.65	25.60
PET	55.63	42.25	18.12	53.18	53.50	22.11	68.39	60.28	21.19
Prompt-based FT (manual)	58.20	57.40	33.40	59.30	62.20	29.20	68.40	63.80	24.20
+ random_demo	60.35	53.20	21.80	60.60	62.80	26.63	72.80	61.80	26.78
+ sim_demo	54.60	54.20	22.80	57.20	55.81	25.22	68.00	64.12	24.20
+ ours	67.80	62.50	36.40	68.14	65.76	33.20	75.12	73.48	34.90
Prompt-based FT (automatic)	63.80	51.42	21.00	65.20	56.00	22.20	71.10	65.80	22.80
+ random_demo	63.20	43.20	21.61	63.00	43.70	26.00	72.32	62.60	24.41
+ sim_demo	62.60	34.80	20.64	64.20	42.20	24.20	69.80	64.00	23.82
+ ours	67.80	57.60	37.63	71.60	66.20	33.00	74.94	66.60	30.80
Fine-tuning (full)**	81.07**								

Table 4: Accuracy on empathy dataset (i.e., iEmpathize) using RoBERTa-base. For the definitions refer to the Table 2.

Model	K = 4			K = 8			K = 16		
	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL	MAIN	NOISYS	NOISYL
					MELD				
Ours (manual)	48.68	37.10	18.80	52.37	39.92	29.91	47.72	42.26	18.85
- MoEx	45.74	35.86	17.06	51.03	38.74	20.57	45.62	39.55	17.69
- CBDemo	40.26	27.56	15.31	46.74	31.50	19.40	44.98	37.64	14.36
Ours (automatic)	43.70	41.90	19.90	47.80	38.95	24.05	50.25	41.05	20.32
- MoEx	40.68	39.34	18.25	46.61	38.25	22.89	48.61	39.51	18.96
- CBDemo	40.30	32.47	17.94	42.45	34.80	19.22	45.34	34.77	15.97
					GoodNewsEveryone				
Ours (manual)	25.75	28.98	24.45	31.50	30.12	29.02	34.87	32.12	29.98
- MoEx	24.68	27.42	22.34	28.44	27.76	27.42	33.80	30.02	26.48
- CBDemo	24.30	20.36	16.60	26.43	19.70	22.55	31.19	26.60	23.70
Ours (automatic)	31.50	27.05	24.58	30.10	28.75	26.90	35.69	33.21	31.46
- MoEx	28.83	24.84	23.78	28.56	27.96	24.42	34.58	30.48	29.50
- CBDemo	28.70	20.34	18.40	27.30	18.82	18.43	32.28	22.70	23.14
					iEmpathize				
Ours (manual)	67.80	62.50	36.40	68.14	65.76	33.20	75.12	73.48	34.90
- MoEx	62.80	61.20	35.60	63.40	63.45	32.96	73.60	71.40	33.68
- CBDemo	63.24	58.55	29.40	64.25	63.10	28.80	73.53	66.49	30.12
Ours (automatic)	67.80	57.60	37.63	71.60	66.20	33.00	74.94	66.60	30.80
- MoEx	64.60	56.00	36.85	66.40	64.55	28.45	73.96	64.20	28.30
- CBDemo	64.67	48.50	25.42	65.49	48.50	28.20	73.20	64.29	27.89

Table 5: Ablation study to investigate the impact of each component in our proposed prompt-based fine-tuning. We report results (% Accuracy) of our approach without using MoEx (i.e., -MoEx), and without using CBDemo (i.e., -CBDemo)

ing cartography-based demonstrations in context through CBDemo (see ours in the tables) results in constant improvement over all the few-shot settings. For example, on MELD with $K = 4$ and manual templates, our proposed approach increased the performance by 7.75% compared to random_demo and by 5.31% compared to the sim_demo.

Interestingly, we also observe that our proposed approach outperforms other baselines with noisy samples, which shows the robustness and effectiveness of our proposed strategy. We can see from the tables that without MoEx data augmentation and by randomly or solely concatenating similar examples (see random_demo or sim_demo in the tables) to the input

sample as demonstrations when the input labels are erroneous (i.e., NOSIYL), the accuracy is significantly degraded compared to the prompt-based fine-tuning (i.e., Prompt-based FT) in most of the cases. Such an aggravation of performance is potentially due to the misleading behavior of these samples that is exacerbated by using similar demonstrations as most hard-to-learn samples are challenging for the model. We can observe similar behavior by adding noise to the input samples in the NOISYS setting. These results suggest that prompting language models by exchanging moments between the training samples along with the cartography-based demonstrations as input context can be robustly used in real-world scenarios where the samples or annotations are noisy.

From the tables, we can also observe that, in comparison with the GPT-3 in-context learning, our prompt-based zero-shot prediction achieves much better performance. The results suggest that using 32 randomly selected demonstrations from different classes with smaller language models (i.e., RoBERTa) is not as effective as in larger models like GPT-3. It is apparent from tables that fine-tuning using the full training set results in an increase in performance.

Ablation Study

We examine the impact of each component (i.e., MoEx and CBDemo) and ablate them in our proposed approach. As shown in Table 5, our proposed prompt-based fine-tuning without the MoEx (i.e., -MoEx) and without cartography demonstrations (i.e., -CBDemo) negatively impact the performance. In our method, without MoEx, we use the training data without exchanging moments between samples and leverage training dynamics to generate informative demonstrations as input context. In our method without CBDemo, we conduct the moment exchange between samples during training and randomly pick one sample from each class for generating demonstrations as input context. The results demonstrate that both components (MoEx and CBDemo) are required to enhance model performance.

Conclusion

In this work, we proposed a novel data-agnostic strategy for prompt-based fine-tuning that leverages feature moments as a data augmentation technique and employs training dynamics to allow more informative samples to be concatenated for generating demonstrations as input context. Our approach is a strong method for few-shot learning that forces the language model to pay special attention to the feature moments and allows more informative samples to be concatenated for generating demonstrations as input context by selecting high confidence and low variance samples. We empirically validate that our method not only achieves the best performance with only a few numbers of labeled samples compared to the other methods but also can be robustly used in real-world scenarios where the samples or annotations are noisy.

Acknowledgements

This research is supported in part by NSF Convergence Accelerator award #2137846, NSF IIS award #2107487, and NSF BigData award #1912887. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF. We thank AWS for computational resources that supported this work. We also thank our anonymous reviewers for their constructive feedback.

References

Bostan, L. A. M.; Kim, E.; and Klinger, R. 2020. GoodNewsEveryone: A Corpus of News Headlines Annotated with Emotions, Semantic Roles, and Reader Perception. In Calzolari, N.; Béchet, F.; Blache, P.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Isahara, H.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, 1554–1566. European Language Resources Association.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Davison, J.; Feldman, J.; and Rush, A. M. 2019. Commonsense Knowledge Mining from Pretrained Models. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 1173–1178. Association for Computational Linguistics.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Ekman, P. 1992. Are there basic emotions? *American Psychological Association*, abs/3.550.553.

Gao, T.; Fisch, A.; and Chen, D. 2021. Making Pre-trained Language Models Better Few-shot Learners. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 3816–3830. Association for Computational Linguistics.

Hosseini, M.; and Caragea, C. 2021a. Distilling Knowledge for Empathy Detection. In Moens, M.; Huang, X.; Specia,

L.; and Yih, S. W., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, 3713–3724. Association for Computational Linguistics.

Hosseini, M.; and Caragea, C. 2021b. It Takes Two to Empathize: One to Seek and One to Provide. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 13018–13026. AAAI Press.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Bach, F. R.; and Blei, D. M., eds., *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, 448–456. JMLR.org.

Jiang, Z.; Xu, F. F.; Araki, J.; and Neubig, G. 2020. How Can We Know What Language Models Know. *Trans. Assoc. Comput. Linguistics*, 8: 423–438.

Li, B.; Wu, F.; Lim, S.; Belongie, S. J.; and Weinberger, K. Q. 2021. On Feature Normalization and Data Augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 12383–12392. Computer Vision Foundation / IEEE.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019a. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P. S. H.; Bakhtin, A.; Wu, Y.; and Miller, A. H. 2019. Language Models as Knowledge Bases? In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2463–2473. Association for Computational Linguistics.

Poria, S.; Hazarika, D.; Majumder, N.; Naik, G.; Cambria, E.; and Mihalcea, R. 2019. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 527–536. Association for Computational Linguistics.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*, 12.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 3980–3990. Association for Computational Linguistics.

Schick, T.; Schmid, H.; and Schütze, H. 2020. Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. In Scott, D.; Bel, N.; and Zong, C., eds., *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, 5569–5578. International Committee on Computational Linguistics.

Schick, T.; and Schütze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 255–269. Association for Computational Linguistics.

Swayamdipta, S.; Schwartz, R.; Lourie, N.; Wang, Y.; Hajishirzi, H.; Smith, N. A.; and Choi, Y. 2020. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of EMNLP 2020, Online, November 16-20, 2020*, 9275–9293. Association for Computational Linguistics.

Talmor, A.; Elazar, Y.; Goldberg, Y.; and Berant, J. 2020. oLMpics - On what Language Model Pre-training Captures. *Trans. Assoc. Comput. Linguistics*, 8: 743–758.

Wei, J. W.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 6381–6387. Association for Computational Linguistics.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Conditional Prompt Learning for Vision-Language Models. *CoRR*, abs/2203.05557.

Zhu, X.; Zhu, J.; Li, H.; Wu, X.; Wang, X.; Li, H.; Wang, X.; and Dai, J. 2021. Uni-Perceiver: Pre-training Unified Architecture for Generic Perception for Zero-shot and Few-shot Tasks. *CoRR*, abs/2112.01522.