

Prompt-based Fine-tuning for Emotion Recognition in Conversation

Zikai Chen 鶴岡 慶雅
東京大学大学院

{chen, tsuruoka}@logos.t.u-tokyo.ac.jp

Abstract

Emotion Recognition in Conversation (ERC) is an important task in Natural Language Processing. However, few studies on ERC have fully exploited the knowledge within pre-trained language models (PLMs), or investigated ERC in low-resource settings. In this paper, we propose a **P**rompt-based fine-tuning method for **ERC** tasks (**PERC**), which leverages the knowledge of large PLMs to improve the performance of ERC in low-resource settings. We conduct experiments on four widely-used ERC benchmarks and show that our method outperforms or is comparable to current state-of-the-art (SOTA) methods. We also run experiments in the few-shot setting and demonstrate that our method greatly outperforms SOTA baselines.

1 Introduction

Emotion Recognition in Conversation (ERC) is a task aiming at identifying the emotions of each utterance in a dialogue. ERC has garnered growing attention in recent years due to its potential applications in various fields, such as empathetic dialogue systems [1], opinion mining in social media [2], and healthcare [3].

In ERC, the emotions of an utterance are often influenced by the dialogue context. This distinguishes ERC from traditional emotion recognition techniques, which typically operate on individual sentences. Such dependencies can be further classified into two categories: the intra-speaker dependency and the inter-speaker dependency. The intra-speaker dependency refers to the emotion flow within a single speaker, where the current emotion of a speaker may be strongly influenced by his/her previous emotional states. In contrast, the inter-speaker dependency occurs when the emotions of one speaker are affected by other speakers within a conversation. An example of such dependencies

can be seen in Figure 1, which shows a dialogue snippet from the MELD [4] dataset.

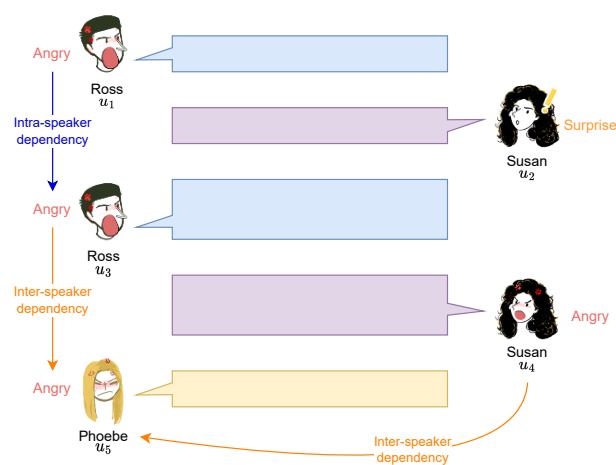


Figure 1 A dialogue snippet from MELD. Intra- and inter-speaker dependencies are represented by blue and orange arrows, respectively.

In previous literature, there has been significant research modeling both intra-speaker and inter-speaker dependencies in ERC with Recurrent Neural Networks (RNNs), Graph Neural Networks (GNNs), or Transformers [5, 6, 7, 8, 9, 10, 11, 12, 13]. However, there are two main limitations on such research. Firstly, many previous studies rely on Pre-trained Language Models (PLMs) to obtain the embeddings of utterances. Some of them directly use PLMs with frozen parameters, which limits the capability of these models. Others fine-tune PLMs during training, but as fine-tuning introduces extra layers on top of PLMs and has a different objective from the pre-training phase, such fine-tuning is prone to overfit the dataset, and knowledge learned during pre-training is not effectively utilized [14].

Secondly, despite the extensive research on ERC, few studies have focused on low-resource settings. This is a critical issue, as it can be challenging to obtain large

dialogue corpora with well-annotated emotion labels. The ability to perform ERC effectively in a low-resource setting is therefore of great importance.

Recently, prompt-based fine-tuning methods have been proposed for various Natural Language Processing (NLP) tasks [15, 16, 17]. In classification tasks, these methods transform the task into a blank-filling problem and predict the labels using a Masked Language Model (MLM) loss function. This aligns with the MLM pre-training objective of PLMs, allowing for better utilization of the knowledge within these models. Prompt-based fine-tuning methods have also shown excellent performance in few-shot learning settings.

We propose a **Prompt-based fine-tuning method for ERC tasks (PERC)**. Our approach involves designing a template that focuses on both intra- and inter-dependencies in dialogues, and using a simple verbalizer to make the process easy to generalize to different ERC tasks.

We conduct experiments on four widely-used ERC datasets, and the results show that our method can achieve better or comparable performance to current state-of-the-art (SOTA) methods. In addition, we evaluate our model in the few-shot setting, and find that it greatly outperforms SOTA baselines.

2 Related Works

2.1 Emotion Recognition in Conversation

ERC has been intensively studied in recent years. In the relatively early stages, techniques such as GRU and LSTM were commonly used to model dependencies between utterances in a dialogue. For example, Wang et al. [6] used general and personalized LSTMs to capture the dependencies within the dialogue. Ghosal et al. [7] employed GRU and common knowledge to update various types of states representing different aspects of the speakers in a conversation. Hu et al. [8] used different LSTMs to obtain the contextual information on each utterance, and then employed a recursive reasoning module to obtain the final representation.

More recently, GNNs and Transformer-based approaches have gained popularity in ERC. For example, Shen et al. [9] built an acyclic graph network to model dependencies among utterances, and fine-tuned a RoBERTa-large model on ERC datasets to obtain utterance representa-

tions. Lee et al. [10] transformed ERC tasks into dialogue-based relation extraction (RE) tasks, and solved these RE tasks with a graph-based network. Liang et al. [11] applied different masks to the sentence-level attention of a Transformer to model various dependencies in a dialogue, and combined this approach with GNNs to model the distance between speakers. Li et al. [12] used a BART-based encoder and used sentence-level attention to model the dependencies. They also combined emotion prediction with contrastive learning and next sentence generation to facilitate training. Lee et al. [5] employed a RoBERTa-large model to encode the entire dialogue context while also maintained a GRU-based memory network for each speaker.

Other tactics that have been proposed for ERC include adding Conditional Random Fields (CRFs) on top of the model layers to facilitate the modeling of the transition of speakers' emotions [6, 11], and using common knowledge from an external knowledge base [7, 13].

2.2 Prompt-based Fine-tuning

Since GPT-3 [18], prompt-based methods have achieved impressive results on a variety of NLP tasks, particularly in few-shot and even zero-shot settings. However, with no gradient updates, such methods require extremely large model sizes to work well, which is not practical in many real-world scenarios [16]. As an alternative, prompt-based fine-tuning has been proposed. By reformulating various tasks as cloze tasks and use an MLM loss function to fine-tune the language model (LM), this approach allows for a smooth transition between pre-training and fine-tuning. Additionally, prompt-based fine-tuning does not introduce any new parameters to the model, making it particularly effective in few-shot settings [15, 16, 17]. In this paper, we apply this approach to the task of emotion classification for each utterance in a dialogue.

3 Method

3.1 Task Definition

In ERC, the dataset D consists of multiple conversations. Each conversation C consists of n utterances $\{u_1, u_2, \dots, u_n\}$, where each utterance u_i is spoken by speaker s_i and annotated with an emotion label y_i from a pre-defined label set Y . In this paper, we consider the



Figure 2 Our proposed prompt template T . x_i and u_i represent speaker s_i 's index and utterance, respectively. In the Memory section, the past utterances of s_i are denoted as $\{u_{m_1}, u_{m_2}, \dots, u_{m_j}\}$, where $m_1 < m_2 < \dots < m_j = i$.

real-time setting, in which the emotion of an utterance u_i is predicted using only the preceding utterances $u_{\leq i}$ and their labels/predictions. This mirrors real-world scenarios, such as emotion prediction in a dialogue system. We only consider the text modal in this paper.

3.2 Prompt Template

We propose a Prompt-based fine-tuning approach for ERC tasks (PERC). First we manually design a template function T that maps each utterance (along with its context information) to its corresponding prompt.

To ensure compatibility across datasets that may not include speaker names, we assign each speaker an index. If speaker s_i is the x -th person to speak in the conversation, we define $x_i = x - 1$. This always makes $x_1 = 0$. Then in the prompt, we simply refer to speaker s_i as Speaker x_i .

As illustrated in Figure 2, our prompt template T consists of three parts: Context, Memory and Question, separated by [SEP] tokens. The Context section includes all the dialogue history up to the current utterance, u_i . For each past utterance u_k , we include Speaker $x_k : u_k$ in the prompt. We expect the LM to learn the inter-speaker dependencies through the Context section.

The Memory section has the same format as the Context section, but only includes the conversation of current speaker s_i . We expect the LM to learn intra-speaker de-

pendencies through the Memory section.

The final part of the template is the Question section, where the LM is asked to predict the emotion of s_i in the utterance u_i . The format of Question is How is Speaker x_i ? [MASK]. Here, Speaker x_i appears again to draw the model's attention to the information related to s_i , and we use a question mark to elicit the answer at the [MASK] position.

3.3 Prompt Answer

In the previous literature, researchers often use a verbalizer V to map label categories to prompt answer words/phrases [15, 19]. Instead of manually design the answer for each label in each dataset, we adopt a simple rule to automatically perform the mapping. For each category label y , we define $V(y)$ as the first sub-word in the tokenized, capitalized label. For example, using the RoBERTa-large language model, $V(\text{anger}) = \text{Anger}$, while $V(\text{sadness}) = \text{Sad}$ because "Sadness" is tokenized into Sad and ##ness. We capitalize the label word first because we place a question mark immediately before the [MASK] token, and we want the prompt to be more natural language-like.

3.4 Training Objective

To fine-tune the model for utterance u_i with label y_i , we maximize the probability of $V(y_i)$ at the [MASK] position with a MLM loss. As there is only one masked position, the MLM loss reduces to a simple cross-entropy loss.

4 Full-shot Experiments

4.1 Datasets

We evaluate our proposed method on four widely-used benchmark datasets: MELD [4], IEMOCAP [20], EmoryNLP (EMORY) [21], and DailyDialog (DD) [22]. Details of these datasets are in Appendix A. Following previous studies, we report the micro-F1 score (ignoring the *neutral* label) on DailyDialog, and the weighted-F1 scores on the other datasets.

4.2 Results

We conduct full-shot experiments with PERC. Experiment details can be found in Appendix B.1. We compare our results with several SOTA baselines, includ-

Models	MELD	IEMOCAP	EMORY	DD
COSMIC	65.21	65.28	38.11	58.48
TODKAT	65.47	61.33	38.69	58.47
DAG-ERC	63.65	68.03	39.02	59.33
CoMPM	66.52	66.33	37.37	60.34
PERC (ours)	66.06	72.53	39.26	60.83
w/o Context	64.70	64.16	37.29	57.32
w/o Memory	66.20	71.58	39.20	60.49
w/ RoBERTa-base	63.57	66.81	36.68	57.80

Table 1 Full-shot results. Results for ablation studies can be found in the bottom of the table.

ing **COSMIC** [7], **TODKAT** [13], **DAG-ERC** [9], and **CoMPM** [5].

From Table 1, we see that **PERC** sets up new SOTA F1 scores on IEMOCAP, EmoryNLP, and DailyDialog. In particular, on IEMOCAP, our performance is 4 points higher than the highest baseline. On MELD, our score is also competitive with the current SOTA, CoMPM.

4.3 Ablation Studies

We also conduct ablation studies on our model. We modify the prompt template by removing the Context section (PERC w/o Context) or the Memory section (PERC w/o Memory). We also investigate the impact of model size by replacing the PLM with the RoBERTa-base model.

The result of ablation studies are shown in the bottom of Table 1. In all cases except for PERC w/o Memory on MELD, performances decrease when the corresponding component is removed. Removing the Context section significantly deteriorates the performance on IEMOCAP, indicating the importance of inter-speaker information in this dataset. Removing the Memory section also slightly hurts the performance on all datasets except MELD. The reason why the performance increases on MELD when the Memory section is removed may be that the MELD is taken from the script of the TV show *Friends*, and a conversation may contain multiple scenes, resulting in incoherence in the same speaker’s utterances. As a result, adding the Memory section to the prompt may be ineffective for MELD. As to the model size, decreasing the model size from large to base consistently degrades the performance by 2 to 4 points on all datasets, indicating that the amount of knowledge in the PLM is critical and using a larger model is beneficial.

Models	MELD	IEMOCAP	EMORY	DD
DAG-ERC	16.39	16.53	17.70	9.79
CoMPM	17.30	35.25	12.82	7.61
PERC (ours)	38.08	51.51	21.79	28.90

Table 2 Few-shot results for $K = 16$.

5 Few-shot Experiments

5.1 Few-shot settings

To create a few-shot setting, following Gao et al. [17], we assume that for each class in the dataset, only K instances are available in the train set and the development set, respectively. Each instance consists of an utterance with its speaker information and emotion label annotation, as well as its full context (all preceding utterances and speaker information), but the emotional labels for preceding utterances are not provided.

5.2 Results

We use the datasets described in Section 4.1. Experiment details can be found in Appendix B.2. We conduct few-shot experiments with **DAG-ERC**, **CoMPM**, and our **PERC**. The results for $K = 16$ are presented in Table 2. Our method achieves significantly better results than the baselines on all datasets.

6 Conclusion

In this paper, we proposed a prompt-based fine-tuning method for Emotion Recognition in Conversation. By designing a template that focuses on both intra- and inter-speaker dependencies and using a simple verbalizer to map emotional labels to prompt answers, our method is able to achieve better or comparable performance to current state-of-the-art methods on four widely-used datasets. In addition, our method also greatly outperforms baselines in the few-shot setting.

References

- [1] Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. MIMe: MIMicking emotions for empathetic response generation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 8968–8979, Online, November 2020. Association for Computational Linguistics.
- [2] Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In **Proceedings of the 13th International Workshop on Semantic Evaluation**, pp. 39–48, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.
- [3] Francisco A. Pujol, Higinio Mora, and Ana Martínez. Emotion recognition to improve e-healthcare systems in smart cities. In Anna Visvizi and Miltiadis D. Lytras, editors, **Research & Innovation Forum 2019**, pp. 245–254, Cham, 2019. Springer International Publishing.
- [4] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 527–536, Florence, Italy, July 2019. Association for Computational Linguistics.
- [5] Joosung Lee and Woojin Lee. CoMPM: Context Modeling with Speaker’s Pre-trained Memory Tracking for Emotion Recognition in Conversation, April 2022. arXiv:2108.11626 [cs] version: 3.
- [6] Yan Wang, Jiayu Zhang, Jun Ma, Shaojun Wang, and Jing Xiao. Contextualized Emotion Recognition in Conversation as Sequence Tagging. p. 10.
- [7] Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. COSMIC: COMmonSense knowledge for eMotion Identification in Conversations, October 2020. arXiv:2010.02795 [cs].
- [8] Dou Hu, Lingwei Wei, and Xiaoyong Huai. DialogueCRN: Contextual Reasoning Networks for Emotion Recognition in Conversations. **arXiv:2106.01978 [cs]**, June 2021. arXiv: 2106.01978.
- [9] Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. Directed Acyclic Graph Network for Conversational Emotion Recognition. **arXiv:2105.12907 [cs]**, September 2021. arXiv: 2105.12907.
- [10] Bongseok Lee and Yong Suk Choi. Graph Based Network with Contextualized Representations of Turns in Dialogue. **arXiv:2109.04008 [cs]**, September 2021. arXiv: 2109.04008.
- [11] Chen Liang, Chong Yang, Jing Xu, Juyang Huang, Yongliang Wang, and Yang Dong. S+PAGE: A Speaker and Position-Aware Graph Neural Network Model for Emotion Recognition in Conversation, December 2021. arXiv:2112.12389 [cs, eess].
- [12] Shimin Li, Hang Yan, and Xipeng Qiu. Contrast and Generation Make BART a Good Dialogue Emotion Recognizer, January 2022. arXiv:2112.11202 [cs] version: 2.
- [13] Lixing Zhu, Gabriele Pergola, Lin Gui, Deyu Zhou, and Yulan He. Topic-Driven and Knowledge-Aware Transformer for Dialogue Emotion Detection. **arXiv:2106.01071 [cs]**, June 2021. arXiv: 2106.01071.
- [14] Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction. In **Proceedings of the ACM Web Conference 2022**, pp. 2778–2788, April 2022. arXiv:2104.07650 [cs].
- [15] Timo Schick and Hinrich Schütze. Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference, January 2021. arXiv:2001.07676 [cs].
- [16] Timo Schick and Hinrich Schütze. It’s Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners, April 2021. arXiv:2009.07118 [cs].
- [17] Tianyu Gao, Adam Fisch, and Danqi Chen. 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)**, pp. 3816–3830, Online, August 2021. Association for Computational Linguistics.
- [18] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Nee-lakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. arXiv:2005.14165 [cs].
- [19] Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. PPT: Pre-trained prompt tuning for few-shot learning. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 8410–8423, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [20] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeanette Chang, Sungbok Lee, and Shrikanth Narayanan. Iemocap: Interactive emotional dyadic motion capture database. **Language Resources and Evaluation**, Vol. 42, pp. 335–359, 12 2008.
- [21] Sayyed M. Zahiri and Jinho D. Choi. Emotion detection on tv show transcripts with sequence-based convolutional neural networks, 2017.
- [22] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset, 2017.

A Dataset Details

Statistics of MELD, IEMOCAP, EmoryNLP (EMORY), and DailyDialog (DD) are shown in Table 3.

	MELD	IEMOCAP	EMORY	DD
#Dlg	1,432	151	897	13,118
Train	1,038	108	713	11,118
Dev	114	12	99	1,000
Test	280	31	85	1,000
#Utt	13,708	7,380	12,606	102,979
Train	9,989	5,154	9,934	87,170
Dev	1,109	604	1,344	8,069
Test	2,610	1,622	1,328	7,740

Table 3 Statistics of ERC datasets. Numbers of dialogues (#Dlg) and utterances (#Utt) in train/dev/test sets are listed.

MELD: The Multimodal EmotionLines Dataset is a multimodal dataset collected from the TV show *Friends*. In our experiments, we only use the text modality. 7 emotion labels are included: *neutral, joy, surprise, sadness, anger, disgust, and fear*.

IEMOCAP: The Interactive Emotional Dyadic Motion Capture is a multimodal dataset collected from a scripted dialogue between two actors. In our experiments, we only use the text modality. It contains 6 emotion labels: *neutral, happiness, sadness, anger, frustration, and excitement*.

EmoryNLP (EMORY): The EmoryNLP dataset is also collected from the TV show *Friends*, but it only includes the text modality. The dataset uses a different emotion set: *mad, scared, neutral, joyful, sad, peaceful, and powerful*.

DailyDialog (DD): The DailyDialog corpus consists of communications from English learners. This dataset includes 7 emotion labels: *neutral, happiness, surprise, sadness, anger, disgust, and fear*.

B Experiment Details

B.1 Full-shot Experiments

In full-shot experiments, we use the RoBERTa-large as our PLM and `</s></s>` as the [SEP] token accordingly. We use the AdamW optimizer and a linear learning rate schedule strategy, with the maximum learning rate set to $1e-5$. We train for 2 to 5 epochs depending on the dataset size and convergence speed, with 1 epoch for warmup. The batch size is 8, with gradient being accumulated every 4 steps, resulting in an equivalent batch size of 32. We run

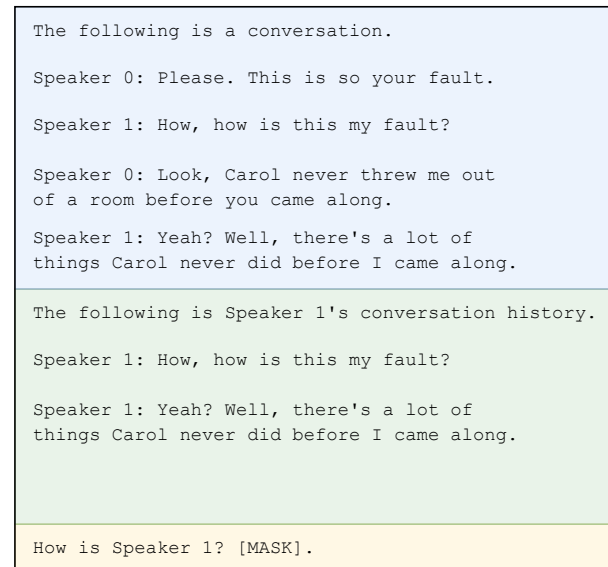
the experiment on 3 random seeds and report the average scores.

B.2 Few-shot Experiments

For each dataset, we first sample K instances for the train set and the development set for each class, and then run the same training process as in Section B.1. Finally, we evaluate our model on the full test set. Because few-shot training can be unstable, we repeat the sample, training, and evaluation process 5 times and report the average score.

C Prompt Example

Figure 3 shows an example of the emotion recognition for u_4 in Figure 1.



```
The following is a conversation.

Speaker 0: Please. This is so your fault.

Speaker 1: How, how is this my fault?

Speaker 0: Look, Carol never threw me out
of a room before you came along.

Speaker 1: Yeah? Well, there's a lot of
things Carol never did before I came along.

The following is Speaker 1's conversation history.

Speaker 1: How, how is this my fault?

Speaker 1: Yeah? Well, there's a lot of
things Carol never did before I came along.

How is Speaker 1? [MASK].
```

Figure 3 A prompt example of the emotion recognition for u_4 in Figure 1.