Emotion-Conditioned Text Generation through Automatic Prompt Optimization

Yarik Menchaca Resendiz and **Roman Klinger**
Institut für Maschinelle Sprachverarbeitung, University of Stuttgart
{yarik.menchaca-resendiz,roman.klinger}@ims.uni-stuttgart.de

Abstract

Conditional natural language generation methods often require either expensive fine-tuning or training a large language model from scratch. Both are unlikely to lead to good results without a substantial amount of data and computational resources. Prompt learning without changing the parameters of a large language model presents a promising alternative. It is a cost-effective approach, while still achieving competitive results. While this procedure is now established for zero- and few-shot text classification and structured prediction, it has received limited attention in conditional text generation. We present the first automatic prompt optimization approach for emotion-conditioned text generation with instruction-fine-tuned models. Our method uses an iterative optimization procedure that changes the prompt by adding, removing, or replacing tokens. As objective function, we only require a text classifier that measures the realization of the conditional variable in the generated text. We evaluate the method on emotion-conditioned text generation with a focus on event reports and compare it to manually designed prompts that also act as the seed for the optimization procedure. The optimized prompts achieve 0.75 macro-average F_1 to fulfill the emotion condition in contrast to manually designed seed prompts with only 0.22 macro-average F_1.

1 Introduction

Emotions are fundamental in communication, where they play an important role in transferring meaning and intent (Ekman, 1992). Emotion-conditioned natural language generation models aim at improving human–computer interaction, by generating text that is not limited to conveying propositional information. However, state-of-the-art conditional generation models require a large amount of data and computational power to achieve models that allow for a fine-grained control over the generated texts (Pascual et al., 2021; Ghosh et al., 2017; Song et al., 2019; Zhou et al., 2018; Menchaca Resendiz and Klinger, 2023).

I.	Input prompt	Generated text
0	`Text with disgust`	`Disgust is a character in Inside Out`
1	`Text expressing disgust`	`Disgusting`
2	`Write a text to express disgust`	`A look of disgust came over his face.`

Table 1: Hypothetical example for a prompt optimization process. The seed prompt is given in Iteration (I.) 0 and misinterpreted to mention the character "Disgust". This issue is fixed through iterative optimization.

In areas like text classification or structured prediction, prompt optimization has established itself as a zero- or few-shot learning paradigm (Ding et al., 2022; Zhang et al., 2022; Wang et al., 2022), also in emotion analysis (Plaza-del Arco et al., 2022; Zheng et al., 2022; Yin et al., 2019). Here, only parameters that are concatenated to the input are optimized and the large language model's parameters are frozen. Such models, therefore, exploit encoded knowledge in models such as Flan (Tay et al., 2023), GPT-3 (Brown et al., 2020) and Alpaca (Taori et al., 2023) more explicitly than fine-tuning them for the task at hand. The optimization method learns "how to use" a model, not "how to change" it.

In recent instruction-based models, the prompt is an instruction to elicit a desired response. The instruction serves as a starting point for generating text that aligns with the intended task. Prompting in text classification (Hu et al., 2022; Gu et al., 2022) usually includes the instruction (e.g., "classify the text…") and the label representation (e.g., "positive", "negative"). Summarization has been represented as an instruction by appending "TL;DR" or "summarize" (Radford et al., 2019; Narayan et al., 2021). For text generation that translates tables to text, Li and Liang (2021) proposed to tune a prefix prompt to accomplish the task. In machine translation, prompts typically mention the source and target language, such as “translate English to German” (Raffel et al., 2020).

The task of prompt optimization can be formulated in various directions. The goal is to find the optimal sequence of tokens to represent the prompt for a specific model (e.g., Flan) and task (e.g., summarization), while keeping the model weights unchanged. AutoPrompt (Shin et al., 2020) defines the prompt optimization as “fill-in-the-blanks” based on a gradient-guided search. OpenPrompt (Ding et al., 2022) provides a toolkit for training prompts using a template dataset, along with corresponding verbalizers for different classes. Deng et al. (2022) use reinforcement learning to infer a successful prompt variation strategy. A different approach for optimization is fine-tuning the model to improve its performance with a specific prompt, while keeping the prompt unchanged (Jian et al., 2022; Gu et al., 2022).

In contrast to most previous work, we use models that have been fine-tuned to solve instruction-based tasks; in our case to generate emotion-conditioned texts. This comes with distinct challenges because the loss function cannot be determined by a single expected label (e.g., positive or negative). In our work, we use a classifier that measures the fulfillment of the condition as a source to calculate the value of an objective function. The optimization procedure that we propose is an evolutionary optimization method (Simon, 2013). Next to the objective function, an important component are actions that allow changes to a prompt to explore the search space.

2 Methods

We propose a method (summarized in pseudocode in Algorithm 1) for text generation conditioned on emotions using prompt optimization. It involves an iterative optimization procedure with three modules, namely *prompt modification*, *text generation*, and *prompt evaluation*. We describe the modules in Section 2.1 and the iterative optimization in Section 2.2.

2.1 Modules

Prompt modification. In each optimization iteration, we apply the three operations, one at a time, to all the tokens in the prompt. Therefore, based on one “parent” prompt, we create $\lambda > 1$ “children”.

Original Prompt	Oper.	Modified Prompt
Text that expresses	Add.	Text <u>string</u> that expresses
Text **that** expresses	Repl.	Text <u>a</u> expresses
Text **that** expresses	Rem.	Text expresses

Table 2: The prompt operations (Oper.) are performed on the same prompt. The Addition (Add.) adds RoBERTa’s special mask token (<mask>) between *Text* and *that*. The Replacement (Repl.) masks the target word (that). The unmasked/predicted tokens by RoBERTa are <u>underlined</u>, and the replaced or removed tokens from the original are in **bold**. Removal (Rem.) deletes one token from the prompt.

Addition adds the most probable token at any position within the prompt, including both the beginning and end of the prompt. We use the pre-trained RoBERTa model (Liu et al., 2019) to retrieve probable tokens for each of these positions. *Removal* deletes a token from the prompt. The *Replacement* operation exchanges a token by the most probable token, again as predicted by RoBERTa.

The *Addition* and *Replacement* operations use the <mask> special token to predict the word. We exemplify these operations in Table 2.

Text generation. We then use each of the λ prompt variations to create text using a large pre-trained language model (e.g., Flan). To do so, we instantiate it with the emotion category. We refer to this instantiation as the *Conditional-Prompts*. Each of them consists of the modified prompt and the specified emotion (e.g., “Text that expresses ⟨em⟩”). Here, ⟨em⟩ is replaced by each of the emotion categories under consideration.

Evaluation. Each prompt is then evaluated through the texts that are generated with its instantiated *Conditional-Prompts*. In the evaluation, we do not further consider texts that are a paraphrase of the Conditional-Prompt. We calculate the BLEU score (Papineni et al., 2002) and filter all texts with a score greater than 0.2. For example, a language model could generate “The text expresses joy.” for a Conditional-Prompt “Text that expresses joy”.

The actual evaluation is performed by comparing the emotion condition to the judgment of an emotion classifier, applied to the generated texts. We use the F_1 measure both as an objective function during optimization and for final evaluation. Note that these two scores are based on two separate classifiers, trained on independent data.

2.2 Iterative Optimization

Algorithm 1 shows the iterative prompt optimization for a given seed prompt P (e.g., "Text that expresses"). The optimization is based on a (μ, λ) evolutionary algorithm (Eiben and Smith, 2015), more concretely $(1, \lambda)$, because we keep only the one best-performing prompt for the next optimization iteration. In contrast to a $(\mu+\lambda)$, the respective parent is not further considered in the next iteration. This makes the algorithm less likely to get stuck in a local optimum.

Initially, P_{opt} (the optimized prompt) is initialized with the seed prompt P. Next, each token in P_{opt} is modified using the Addition, Replacement, and Removal. Each operation is performed one at a time, and the results are stored in $\mathbf{P}_{mod}$ (Section 2.1). The *Generate* method produces a text for each *Conditional-Prompt*-combination of the input prompt and the emotion class (e.g., "Text that expresses joy", "Text that expresses anger"; Section 2.1). We compare the generated text from P_{opt} (namely T_{opt}) against the generated text from each modified prompt ($\mathbf{P}_{mod}$), denoted as $\mathbf{T}_{mod}$. If the F_1 of $\mathbf{T}_{mod}$ is higher than that of T_{opt}, the prompt $prompt_{mod}$ is assigned as the new optimized prompt (P_{opt}) and added to the best-performing candidates ($\mathbf{P}_{cands}$). Finally, this process is repeated for a total of N times and P_{opt} is updated with the best-performing prompt from $\mathbf{P}_{cands}$.

Algorithm 1: Automatic Prompt Optimization. *Eval* involves an emotion classifier and the BLEU score.

```
Input  : Seed Prompt P,
         Maximum Iterations N
Output : Optimized Prompt P_opt
P_opt ← P;
i ← 0;
P_cands ← {};
while i < N do
    P_mod ← {};
    for token ∈ P_opt do
        P_mod += Add(P_opt, token);
        P_mod += Replace(P_opt, token);
        P_mod += Remove(P_opt, token);
    T_opt ← {};
    for prompt_mod ∈ P_mod do
        T_mod ← Generate(prompt_mod);
        if Eval(T_mod) > Eval(T_opt) then
            P_opt ← prompt_mod;
            T_opt ← T_mod;
    P_cands += P_opt
    i ← i + 1;
P_opt ← select-one-best(P_cands);
return P_opt;
```

3 Experiments

Section 3.1 explains the experimental settings used to optimize an initial prompt that we assume to be provided by a user. Section 3.3 validates the proposed method by showing that emotion-conditioned text generation improves when using the optimized prompt compared to the seed prompt.

3.1 Experimental Settings

To validate the feasibility of our method for emotion-conditioned text generation, and its cost-effectiveness in terms of data and computational resources, we utilized available pre-trained models and datasets. Specifically, we used Flan (Tay et al., 2023), an open-source model trained on instruction-based datasets, as a generative model. We trained two classifiers using (1) the ISEAR dataset (Scherer and Wallbott, 1994) for prompt optimization in each iteration, and (2) the crowd-enVent dataset (Troiano et al., 2023) for final evaluation, utilizing the same subset of emotions as the ISEAR dataset.[1] Both classifiers are built on top of RoBERTa using default parameters for 10 epochs.[2]

These data sets are independent of each other, and therefore the objective signal is independent of the final evaluation. Both sets, however, are comparable: they contain texts in which people were asked to report on an emotion-triggering event, given a predefined emotion. In the original ISEAR corpus, these texts were acquired in an in-lab setting in the 1990s, while the crowd-enVENT corpus has recently been collected in 2022 in a crowdsourcing setup. An example from the ISEAR corpus is "When I was involved in a traffic accident." – an example from crowd-enVENT is "When my son was poorly with covid".

Prompt Modification. We selected a straightforward seed prompt—"Write a text that expresses ⟨em⟩"—for ten iterations and all operations.

Text Generation. For each *Conditional-Prompt*, we generate the three most probable sentences using a beam search with a beam size of 30, a next-token temperature of 0.7, and a top-p (nucleus)

[1]The emotion labels are: Anger, Disgust, Fear, Guilt, Joy, Sadness, and Shame.

[2]The crowd-enVent and ISEAR-based classifiers have macro-F_1 of .78 and .77, respectively.

I.	Ope.	Optimized Prompt (P_{opt})	F_1
0	—	Write a text that expresses ⟨em⟩	.28
1	Repl.	Write a text to expresses ⟨em⟩	.80
2	Add.	Write in a text to expresses ⟨em⟩	.91
3	Add.	Write in a text string to expresses ⟨em⟩	.88
4	Add.	Write in a long text string to expresses ⟨em⟩	**.94**
5	Rem.	Write in long text string to expresses ⟨em⟩	**.94**
6	Repl.	Write in long text strings to expresses ⟨em⟩	.91

Table 3: Prompt optimization at different iterations (I.), with Iteration 0 representing the seed prompt. The ⟨em⟩ token represents any of the seven emotions in the ISEAR dataset. The macro F_1 score is calculated using the ISEAR classifier, across all the emotions.

sample of 0.7. We ensure that our output excludes sentences with repeated instances of the same bigram.

Prompt Evaluation. We filter out all prompts where the average BLEU score is higher than 0.2 across all the conditional prompts. Next, we select the prompt with the best F_1 score using the ISEAR classifier.

3.2 State-of-the-art Baseline

We compare our method against the plug-and-play method proposed by Pascual et al. (2021)—a state-of-the-art model for affective text generation. To do so, we train the emotion discriminators that are part of that method on top of GPT-2 with the ISEAR dataset. The comparison is not straightforward since this method uses the prompt as a starting point to generate the sentence, whereas our approach treats the prompt as an instruction. Therefore, we select the most frequent n-grams from the ISEAR dataset as prompts: “When I was”, “When a”, and “When someone”. For each prompt–discriminator combination, we generate the 5 most probable sentences.

3.3 Results

We begin the discussion of the results with Table 3, which shows the prompt optimization and performance across iterations. It reveals two notable findings: First, already the first iteration, compared to the seed prompt in Iteration 0, shows an increase by 52 pp in F_1. This is a change only by replacing “that” with “to”. Given our selection criteria which does not include the parent prompt in the selection, there can be a decrease in performance, which can be observed in Iteration 2. Second, all prompts in Table 3—the best-performing prompts at each iteration—are human-readable. This is in contrast to prompt optimization in other NLP tasks, where the resulting prompts often become less human-readable. For example, in the fact retrieval task “[X] limestone depositedati boroughDepending [Y]” performs better than “[X] is the capital of [Y]” (Ding et al., 2022).

Table 4 showcases examples of generated texts from various prompt candidates. The prompt candidates at the same iteration are a few examples of the resulting prompt modifications as described in Section 2. The provided F_1 scores refer to the performance of the prompt across the 7 emotions, not the performance of the specific examples shown. Comparing the generated text from the seed prompt (Row 1) and the first optimization (Row 2), we observe a better fulfillment of the emotion *disgust* for the optimized prompt—the uncertainty expressed in Row 1 indicates *fear*. Prompt modifications at the same iteration have different performances. For example, in Iteration 2 (Rows 4/5), there is a difference of 33 pp in F_1. It is important to note that the best F_1 score does not always indicate an improvement in fulfilling the condition of the generated text. Sometimes, the best-scoring text can be a paraphrase of the prompt, which may be falsely classified as correct due to the presence of the emotion class name (e.g., Row 6/Iteration 5, Row 3/Iteration 2).

Finally, Table 5 shows an independent evaluation of the method along with the results achieved with the method by Pascual et al. (2021). We report F_1 scores for the ISEAR-based classifier used during the optimization process and the independent crowd-enVENT-based classifier. The latter numbers therefore constitute an independent evaluation result. We observe that the numbers of both classifiers are comparable to each other. The comparison to the baseline shows that our seed prompt performs on par with Pascual’s method (.18, .12, and .17 vs. .22, respectively). Our optimized prompt, however, shows a higher performance (.75 F_1).

4 Conclusion and Future Work

In this study, we introduced the first automatic prompt optimization method for text generation conditioned on emotions. Our approach involved three token operations: addition, replacement, and removal. We utilized a BLEU score and an automatic classifier to filter and rank the modified prompts. We demonstrated that the optimized prompts led to a higher fulfillment of the intended

Iter.	Prompt	Generated text	F_1
0	Write a text that expresses disgust	i don't know what to do about it.	.28
1	Write a text to expresses disgust	i'm disgusted.	.80
1	a text that expresses fear	a text that expresses fear	.90
2	Write in a text to expresses shame	i don't know how to express my shame.	.91
2	Write a text that to expresses shame	i'm sorry to hear that.	.58
5	Write in a long enough string to expresses joy	a long enough string to express joy.	1.0
5	Write a long text string to expresses joy	i love you so much	**.94**

Table 4: Generated text examples from the same seed prompt (1st row) at different optimization steps. The macro F_1 score is computed for the prompt across all the emotions using the ISEAR classifier.

Method	Prompt	ISEAR	crowd-enVent
Pascual (2021)	When I was	.18	.18
	When a	.43	.12
	When someone	.21	.17
P_{opt}	Write a text that expresses ⟨em⟩	.28	.22
	Write in long text string to expresses ⟨em⟩	.94	**.75**

Table 5: Comparison between our method (P_{opt}) and Pascual (2021) Rows 1–3 are the most frequent n-grams for the ISEAR dataset. The 4th row corresponds to the seed prompt, and the 5th row represents the optimized prompt. The macro-average F_1-score for both ISEAR and crowd-enVent datasets is computed across all emotions.

emotions compared to the seed prompt, with a 53 pp improvement in the F_1 score. It is a cost-effective method in terms of both data and resource requirements, while still achieving good results.

This leads to important future work. While our approach improves emotion-conditioned text generation, there are several areas that need to be explored further. First, we need to explore different search techniques for prompt optimization (e.g., Beam search). Second, it is essential to compare the performance of the optimized prompts across different domains to assess the generalizability of our method. Our evaluation is arguably comparably narrow, with only one seed prompt and one domain in which emotions are expressed. Finally, it is crucial to analyze our approach by comparing it against a fine-tuned or trained model from scratch to evaluate its effectiveness and efficiency.

Another interesting direction of research would be to study in more detail how the expected domain of the generated texts (here: emotion self-reports) might be in conflict with the emotion condition and how that can be encoded in either the optimization process, the seed prompt selection or the objective functions, or in combinations of these parameters.

5 Ethical Considerations & Limitations

The proposed method aims at optimizing prompts for conditional text generation, particularly when conditioned on emotions. The generated affective texts do not only serve as a source to study the capabilities of large language models from a computational perspective. We believe that they can also be of value to better understand the representation of psychological concepts in automatically generated text. However, there are some risks associated with the method if not used with care, primarily inherited from the underlying language model. Optimized prompts could potentially result in generating text that reinforces stereotypes or marginalize certain groups. When dealing with the expression of emotions, it is essential to exercise caution when employing these models due to their potential impact on individuals.

A limitation in our evaluation and method is that we rely heavily on the seed prompts. This can lead to fast convergence—if the seed prompt is adequate for the task, the optimization process is more likely to be successful. The optimization is based on a (μ, λ) approach, which can be seen as a brute-force search. However, alternative search algorithms may provide a more efficient optimization of the prompt in terms of iterations.

Overall, the method has proven to be useful for text generation conditioned on emotions. We invite people to keep the above limitations in mind when considering the capabilities and applications of our method.

Acknowledgements

This work has been supported by a CONACYT scholarship (2020-000009-01EXTF-00195) and by the German Research Council (DFG), project "Computational Event Analysis based on Appraisal Theories for Emotion Analysis" (CEAT, project number KL 2869/1-2).

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. OpenPrompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland. Association for Computational Linguistics.

Agoston E. Eiben and James E. Smith. 2015. *Introduction to evolutionary computing*. Springer.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-LM: A neural language model for customizable affective text generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 634–642, Vancouver, Canada. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. 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)*, pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 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)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

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

Xiang Lisa Li and Percy Liang. 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)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Yarik Menchaca Resendiz and Roman Klinger. 2023. Affective natural language generation of event descriptions through fine-grained appraisal conditions. In *Proceedings of the 16th International Conference on Natural Language Generation*, Prague, Czech Republic. Association for Computational Linguistics.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Flor Miriam Plaza-del Arco, María-Teresa Martín-Valdivia, and Roman Klinger. 2022. Natural language inference prompts for zero-shot emotion classification in text across corpora. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6805–6817, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Klaus R. Scherer and Harald G. Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Dan Simon. 2013. *Evolutionary Optimization Algorithms*. John Wiley & Sons, Hoboken, New Jersey, USA.

Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuanjing Huang. 2019. Generating responses with a specific emotion in dialog. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3695, Florence, Italy. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Enrica Troiano, Laura Oberländer, and Roman Klinger. 2023. Dimensional modeling of emotions in text with appraisal theories: Corpus creation, annotation reliability, and prediction. *Computational Linguistics*, 49(1).

Jianing Wang, Chengyu Wang, Fuli Luo, Chuanqi Tan, Minghui Qiu, Fei Yang, Qiuhui Shi, Songfang Huang, and Ming Gao. 2022. Towards unified prompt tuning for few-shot text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 524–536, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Haoxing Zhang, Xiaofeng Zhang, Haibo Huang, and Lei Yu. 2022. Prompt-based meta-learning for few-shot text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1357, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaopeng Zheng, Zhiyue Liu, Zizhen Zhang, Zhaoyang Wang, and Jiahai Wang. 2022. UECA-prompt: Universal prompt for emotion cause analysis. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7031–7041, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.