SemEval-2023 Task 4: Fine-tuning vs Prompting, Can Language Models Understand Human Values?

Pingwei SUN
BDT program, HKUST
`psunah@connect.ust.hk`

Abstract

Accurately handling the underlying support values in sentences is crucial for understanding the speaker's tendencies, yet it poses a challenging task in natural language understanding (NLU). In this article, we explore the potential of fine-tuning and prompt tuning in this downstream task, using the *Human Value Detection 2023*. Additionally, we attempt to validate whether models can effectively solve the problem based on the knowledge acquired during the pre-training stage. Simultaneously, our interest lies in the capabilities of large language models (LLMs) aligned with RLHF in this task, and some preliminary attempts are presented.

1 Introduction

The persuasiveness of arguments is heavily influenced by individuals' values, and variations in the priority assigned to these values can lead to disputes, especially between different cultures. To address this, computational linguistics leverages human values to categorize and evaluate arguments.

Schwartz's value categorization(Schwartz, 1994) serves as a common framework. However, existing datasets have limitations, such as small sizes and cultural biases. To overcome these issues, the competition holder proposes an extension, Touché23-ValueEval (Mirzakhmedova et al., 2023), containing 9324 diverse arguments from various sources and cultures. This dataset aims to facilitate the development of Pre-trained Language Models (PLM) for the automatic identification of human values in persuasive communication, supporting a broad array of applications. The task attracted many researchers to work on it and various methods were proposed to solve the problem.

However, from the leaderboard, this task still requires more in-depth research, with the top performer only achieving an average F1 score of 0.56. Most participating teams tried to solve it as a classification task and employed techniques such as model ensembling and expanding the dataset to improve scores. Taking into account computational resources and the rise of LLMs, our project will focus on the following questions based on the dataset:

- **Q1:** Does prompt-tuning work compared with fine-tuning on the complex downstream task?

- **Q2:** Can PLMs handle human values powered by knowledge from the pre-training stage?

- **Q3:** How do LLMs perform on the task after aligning with human preference?

2 Related Work

Human Value Detection 2023. Among the submissions, most teams applied transformers-based models as their backbone and treated it as a classification task.

Team Adam Smith (Schroter et al., 2023) employs an ensemble strategy, including DeBERTa and RoBERTa. They trained these models for loss minimization or F1-score maximization on three folds each RoBERTa model was pretrained on the IBM-ArgQ-Rank30KArgs dataset. Ensembling involved averaging predictions with an optimized decision threshold. They experimented with a stacked meta-classifier based on logistic regression.

Team John Arthur (Balikas, 2023) fine-tuned a DeBERTa model on the task's data, using a concatenated representation of stance, premise, and conclusion. They found that separate token symbols for stance improved classification. The model was trained to minimize binary cross-entropy loss, and the team observed performance benefits with more training data.

Team PAI (Ma, 2023) combined models using various input datasets. They applied weight voting based on F1 score for ensembling and explored different loss functions, ultimately favoring a class-balanced loss. Classification thresholds were tested, but no performance improvement was achieved.

Feature	Value Type	Example
Argument ID	Str	"A01002"
Conclusion	Str	"We should ban fast food"
Stance	Str	["supporting", "against"]
Premise	Str	"Fast food should be banned because it is really bad ..."
Value Labels	Int	[0, 1]
		Value categories: "Achievement", "Stimulation", etc.
Value Description	Str	"Self-direction: thought. It is good to have own ideas and interests.
		Contained values and associated arguments (examples): ..."

Table 1: The dataset used in the experiment with the corresponding types and examples. **Stance** is mapped to single words to simplify the prompting template. Square brackets represent single selections from among. The **Value Description** are defined in Schwartz, 1994.

Fine-tuning. Since the release of BERT (Devlin et al., 2019), the approach of large-scale pre-training followed by fine-tuning on downstream tasks has been widely explored and applied to various NLP subtasks.

Through the integration of a range of training techniques, this method has achieved impressive results. The bi-encoder structure employed during pre-training inherently endows it with the ability to effectively extract semantic information from sequences. However, it performs poorly when facing few-shot and zero-shot scenarios.

Prompting. Recently, researchers have been attempting to transform NLP tasks into a seq-to-seq format using prompts. T5 (Raffel et al., 2020) marked the first successful endeavor in this direction. Such an approach leverages the knowledge acquired by language models during the pre-training phase rather than treating them as feature extractors. Consequently, it exhibits excellent performance on few-shot and zero-shot tasks, with the added benefit of reduced fine-tuning costs.

GPT, with its decoder structure, demonstrates enhanced capabilities in language modeling. The team Hitachi (Tsunokake et al., 2023) has used BART, T5, and GPT3, feeding the data to them in question-answering format.

3 Methods

To answer the questions proposed in § 1, we propose the following techniques, to our best knowledge, to measure the potential of models on the complex task of detecting human values. Figure 1 illustrates our workflow in the project.

3.1 Data Utilization

The data file contains various annotations and supplementary datasets from platforms like Zhihu and The New York Times. Experiments are conducted on the main dataset, and due to the unavailability of test labels, all results in this paper reflect the model's performance on the validation set.

Before commencing the experiments, the data files are reorganized, and certain features are preprocessed, as shown in Table 1.

3.2 Fine-tuning as a Classification Task

Classifier. To solve the multi-label classification task, there are two methods to choose from. On the one hand, it is sensible to use a single classifier, apply the sigmoid function to each x_i in the $output = \{x_1, x_2, ..., x_c\}$, and then calculate the Binary Cross Entropy Loss as follows.

$$\sum_{i=1}^{C} y_i \cdot \log \sigma(x_i) + (1 - y_i) \cdot \log(1 - \sigma(x_i)) \quad (1)$$

On the other hand, it also works by setting up the same number of classifiers as the number of labels, each classifier to handle one category.

Hidden States. Serving as an encoding module, BERT can provide various levels of features of the sequence. It is commonly agreed that the deeper layers always output high-level semantic information (Sun et al., 2019) and the first token [CLS] can be used for sequence classification.

Avoid Forgetting. Since the model has learned solid knowledge during pre-training, we need to find an appropriate way to arrange the learning rate for fine-tuning. The lower layers handle more general information so they should be updated with relatively lower learning rates. We use the decay factor to achieve layer-wise learning rates.
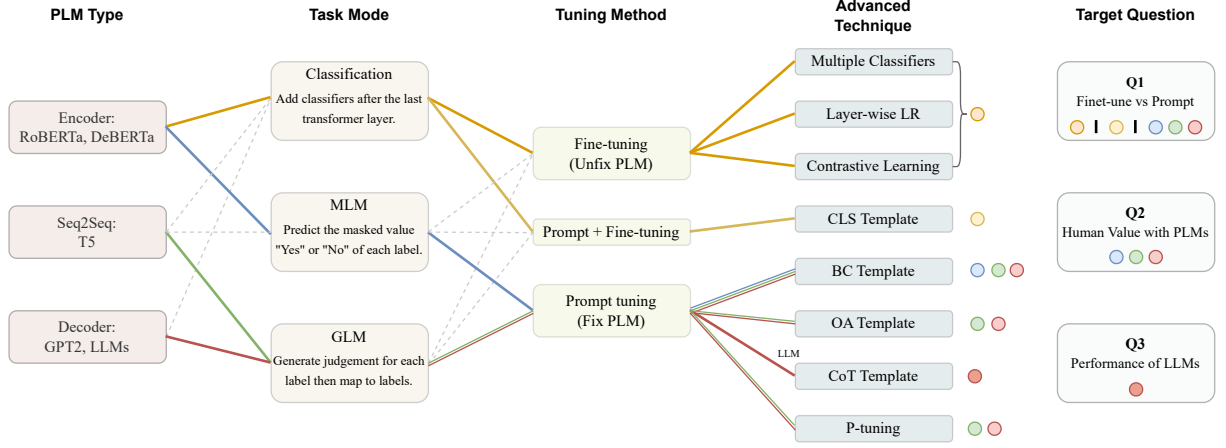
Figure 1: Illustration of the workflow of our experiments. Tracks marked by different colors stand for combinations of models, tasks, and methods, details of which are told in § 4. The dashed lines mean methods are theoretically available but we do not consider them in the project because of poor performance or being computationally demanding. The colorful dots are experiment results and indicate the analysis for the questions proposed in § 1.

3.3 Contrastive Learning

This project is a complex Natural Language Understanding task, so obtaining a good embedding representation of the sequence is crucial for the subsequent training of the classifier.

Inspired by the SimCSE (Gao et al., 2021), Contrastive Learning (CL) loss is applied to optimize the embedding states of sequences. Since the target is multi-label classification, the positive and negative samples can not be defined directly inside a mini-batch. Consequently, we compute the CL loss according to the following formulas:

$$w_{ij} = \frac{y_i^T \cdot y_j}{\sum_{k=1}^{B} y_i^T \cdot y_k + \varepsilon}$$
$$CL\ \ell_i = -\log \frac{\sum_{j=1}^{B} w_{ij} \cdot e^{\frac{sim(x_i,x_j)}{\tau'}}}{\sum_{j=1}^{B} e^{\frac{sim(x_i,x_j)}{\tau'}}} \quad (2)$$

where y is the label vector, x is the embedding representation and B is the batch size. The τ' stands for the temperature hyper-parameter of CL, which is a scaling factor of similarity scores.

3.4 Prompt Tuning Techniques

When doing prompt tuning, the model should be transferred to other modes to fit the template's requirements, shown in Figure 2. The details of template construction are introduced below.

Binary choice template. It is easy to come up with the idea to transfer the multi-label classification task to several binary choice questions. It can be solved either with a Masked Language

Model(MLM) by filling the blank or a Generative Language Model(GLM) by generation.

Open answering template with knowledgeable verbalizer. When our templates become more complex and open, it often allows for better stimulation of internal model knowledge. At this point, we need a verbalizer to map the output results to labels, and the choice of verbalizer significantly impacts the outcomes (Hu et al., 2022).

Notably, the dataset includes descriptions and examples of human values. However, they were artificially defined by linguists two decades ago. To enhance the dataset's generalization, we introduce LLM to rewrite the descriptions and provide synonyms, which will serve as a knowledgeable verbalizer to map the output results to labels of human values.

Chain-of-Thought template. Language models, particularly LLMs, have demonstrated remarkable reasoning capabilities, as evidenced by studies such as (Kojima et al., 2022). Despite the challenge of directly comprehending whether specific human values are implied in triples, models can acquire relevant knowledge through a systematic prompting process. This step-by-step approach allows LLMs to navigate and extract pertinent information, ultimately enabling them to effectively address the intricacies of classification problems.

In light of this, we have incorporated templates in the form of CoT. These templates serve as structured prompts that guide the model in interpreting and understanding the nuances of human values within extra knowledge.

Prompt Template Board

CLS Template

<u>Input:</u>
Human Values are indicated in the premise: **[Premise]** and the conclusion: **[Conclusion] [Stance]** it.
<u>Output:</u>
[Logits for classifiers]

MBC Template

<u>Input:</u>
Given the premise: **[Premise]** and the conclusion: **[Conclusion] [Stance]** it. Is it based on the **[Value Description]**? [MASK]
<u>Output:</u>
"Yes" or "No"

BCA Template

<u>Input:</u>
Question: Given the premise: **[Premise]** and the conclusion: **[Conclusion] [Stance]** it. Is it based on the **[ValueDescription]**? The answer is
<u>Output:</u>
"Yes" or "No"

OA Template

<u>Input:</u>
▢. I claim that **[Conclusion]** because I **[Stance]** the **[Premise]**. In the aspect of **[Value Label]**, use some words to describe what kind of person am I?
<u>Output:</u>
True if any ◯ in the answer, else False.

CoT Template

<u>Input 1:</u>
What kind of human values are indicated in the premise: **[Premise]** and the conclusion: **[Conclusion] [Stance]** it?
<u>Output 1:</u>
#Reply from LLM#
<u>Input 2:</u>
Do you think [#Reply from LLM#] is aligned with ▢, which can also be called ▢?
<u>Output 2:</u>
"Yes" or "No"

- ⬤ Solving all labels in one run
- ◯ Available in encoder with classifier mode
- ◯ Available in MLM mode
- ◯ Available in GLM mode
- ◯ Human value descriptions from ChatGPT
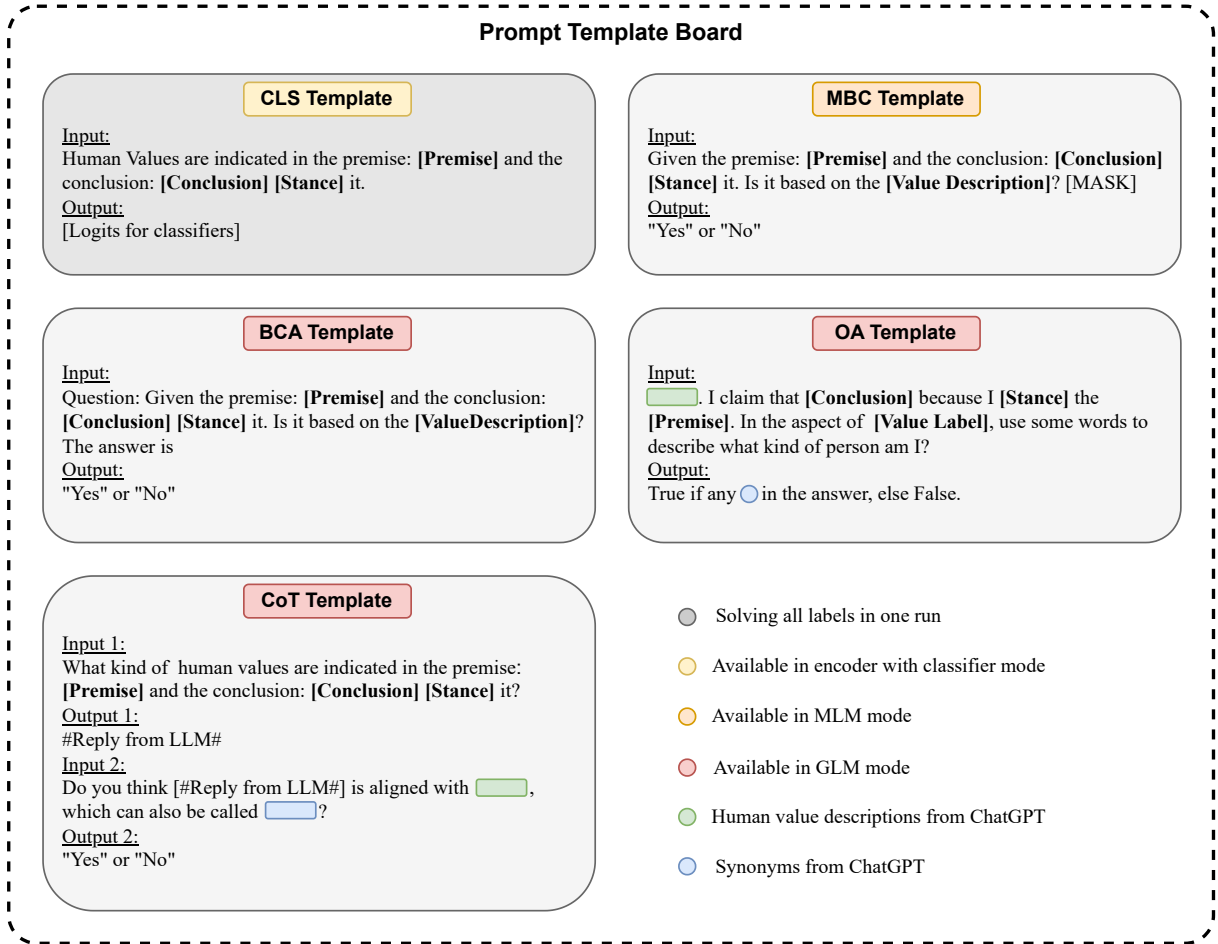- ◯ Synonyms from ChatGPT

Figure 2: Prompt templates for different task processing modes, including classification(CLS), masked binary choice(MBC), binary choice answering(BCA), open answering(OA), and Chain-of-Thought(CoT). The bolded content in brackets represents the features in the dataset, shown in Table 1.

4 Experiments

The following experiments are conducted to answer questions mentioned in § 1. Since we aim to explore fine-tuning methods and have a limited computational budget, tricks such as model ensembling to improve scores are not applied.

4.1 Exp-I: Fine-tuning

Based on the previous team's papers, we choose RoBERTa and DeBERTa followed by a classifier for fine-tuning experiments. The Premise, Conclusion, and Stance features are directly fed to the model. Settings of hyperparameters for training are listed in Table 2.

The performance of different models and classifiers is presented in Table 3. RoBERTa with multiple classifiers is selected after the preliminary fine-tuning experiment.

Then Contrastive Learning is introduced for further improvement of the performance. In this part,

we treat it as either a pre-training task or an extra loss item at the fine-tuning stage. The second strategy performs slightly better, and both of them make progress on the macro F1 score.

Parameters	Value
Trainer	
Batch size	8
Epochs	3.0
Lr scheduler	cosine
Warmup ratio	0.1
Lr decay	0.97
Optimizer	
Optimzier	Adamw
Learning rate	2e-5
Trainable param	last 8 layers + classifier(s)

Table 2: We try various combinations of the hyperparameters and compare validation scores after training. The above is an optimal choice among them.

Model	Macro F1 score
$RoBERTa_{Large}$	
w/ SH	.481
w/ MH	**.507**
$DeBERTa_{Large}$	
w/ SH	.477
w/ MH	.493
$RoBERTa_{Large}$ w/ MH	
CL pre-train	.518
CL fine-tune	**.522**

Table 3: Performance of finetuned encoding models with classifiers on the validation set. SH means single classification head, while MH means multiple heads.

4.2 Exp-II: Prompt Tuning

The previously mentioned prompt templates are sequentially tested in this section in conjunction with models of different parameter sizes and structures. The training strategies are inspired by Liu et al., 2021 and implementations are based on OpenPrompt (Ding et al., 2022).

To be specific, the CLS template is treated as a hard prompt of which parameters are frozen and fed to encoding models with trainable multiple classifiers. Its trainable parameters are consistent with the previous fine-tuning settings.

Templates in the forms of mask filling and question answering are also proposed. At this time, we set PLMs fixed and allow prompts' parameters to be updated. Extra knowledge in the OA template is processed in advance, which is manually selected from feedback from ChatGPT.

4.3 Exp-III: Works on LLMs

The capabilities of the LLMs on this complex NLU task are also what we plan to explore. Experiments are conducted with a CoT template which is designed to stimulate the logical inference ability and knowledge inside LLMs.

Considering the computational budget, no gradient updating process is included in this template. Instead, the scores are obtained by taking 5% of the validation dataset for local inference (Llama7B) or API calls (GPT series) and computing F1 scores on predicted results, which is shown in Table 4.

5 Analysis

5.1 Effectiveness of Fine-tuning and Input Description

We achieved a relatively competitive result compared with last year's leaderboard. At the same time, we also validated the contribution of contrastive learning to obtaining more discriminative embedding representations.

Furthermore, we observed that, in handling NLU tasks with complex inputs, providing reasonable descriptions for complex input examples enhances model performance compared to directly concatenating and inputting individual components.

5.2 Performance of Prompt Tuning

Finding versions of different models with the same parameter size is a challenge, to ensure the validity of the results, we attempted to compensate by the trainable parameters during training.

From the table, it is evident that on models with a size less than 500M, fine-tuning outperforms prompt tuning, which is reasonable being given more trainable parameters. However, when the model size approaches 1B (two to three times that of the formers), prompt tuning shows comparable performance to fine-tuning. Yet, during training, we observed a larger variance in the performance of prompt tuning with changes in the initialization methods of templates.

Does it mean fine-tuning is always a better choice in this range of parameter size ($<$1B)? Our two additional experiments explain this in more detail. The results of NLI and few-shot are listed in Table 5.

We built a simple NLI task using the Premise Conclusion and Stance features in the dataset and tuned some models using the same method as before. As can be seen, there is no significant gap between fine-tuning and prompt tuning in this simpler task, even with fewer trainable parameters, indicating that with the increase in task complexity, prompt tuning requires a larger base model, meaning better language modeling abilities and more knowledge to achieve performance comparable to the encoder + classifier paradigm.

When faced with few-shot datasets, prompt tuning shows obvious advantages. In contrast, fine-tuning seems failed to solve the task.

Method / F1 score	All	Self-direction: thought	Self-direction: action	Stimulation	Hedonism	Achievement	Power: dominance	Power: resources	Face	Security: personal	Security: societal	Tradition	Conformity: rules	Conformity: interpersonal	Humility	Benevolence: caring	Benevolence: dependability	Universalism: concern	Universalism: nature	Universalism: tolerance	Universalism: objectivity
Classification																					
RoBERTa w/ MH	.51	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RoBERTa-CLS	**.54**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DeBERTa-CLS	.52	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MLM																					
RoBERTa-MBC	.47	.52	.63	.22	.31	.58	.33	**.56**	.30	.69	.62	.58	.55	.27	.16	.50	.32	.66	.75	.39	.43
DeBERTa-MBC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GLM																					
T5†-BCA	.45	.49	.59	.23	.30	.58	.37	.50	.25	.70	.61	.44	.46	.25	.21	.48	.28	.68	.73	.33	.50
T5†-OA	.45	.49	.58	**.25**	.31	.58	.32	.49	.27	.73	.60	.49	.45	.25	.19	.47	.28	.66	.74	.35	.52
T5-BCA	.52	**.57**	**.70**	.18	**.44**	**.65**	.41	**.54**	**.33**	**.78**	**.68**	**.67**	**.59**	.30	.13	.50	.35	**.76**	**.85**	**.49**	**.56**
T5-OA	.50	.51	.69	.13	.27	.61	**.44**	.49	.30	.74	.62	.58	.55	**.36**	.17	**.52**	.41	.73	.80	.45	.55
GPT2†-BCA	.42	.41	.57	.15	.21	.55	.26	.43	.23	.66	.53	.47	.49	.31	.20	.42	.28	.64	.72	.36	.53
GPT2†-OA	.44	.45	.62	.11	.42	.47	.31	.41	.23	.71	.56	.44	.51	.33	.20	.45	.30	.67	.77	.39	.52
GPT2-BCA	.46	.44	.63	.19	.23	.58	.38	.47	.25	.69	.56	.52	.54	.34	**.23**	.43	.28	.68	.79	.32	**.56**
GPT2-OA	.48	.47	.65	**.25**	.29	.56	.37	.51	.31	.72	.60	.51	.55	.27	.19	.50	.33	.67	.81	**.52**	.54
LLM-CoT																					
Llama-7B	.49	.55	.66	.20	.39	.64	.35	.48	.26	.77	.63	.54	.53	.35	.20	.53	.38	.71	.80	.34	.42
ChatGPT	.53	.52	.71	.26	.33	.61	.48	.52	.38	.75	.65	.60	.56	.47	.16	.55	.42	.72	.83	.42	.57

Table 4: F1 score of various prompt tuning methods. T5 and GPT2 with † stand for the base and medium versions respectively and the others are of their large versions except for LLMs. There is something wrong with DeBERTaForMaskedLM in huggingface, so DeBERTa-MBC is of no reference value.

5.3 Prompting LLMs to Detect Human Values

In the aspect of LLMs, even without any fine-tuning, a well-designed questioning style (prompts) can enable them to perform well on the task.

However, this improvement is more uncertain than the previously mentioned methods (as can be seen from the variation in scores across categories) and relies on an experienced questioner. Failures in a few categories may be attributed to inconsistencies between external knowledge from ChatGPT and task definitions.

Models / Task	NLI	5-shot HVD
RoBERTa	.872	-
T5	.866	.413
GPT2	.849	.408

Table 5: Results of NLI and few-shot Human Value Detection. The model sizes are close and all below 500M. NLI is evaluated by accuracy while HVD still uses the F1 score.

6 Conclusion

6.1 Fine-tuning vs Prompt Tuning

This is a problem that involves multiple variables as we mentioned in the analysis section. Each approach has its own strengths and weaknesses.

Briefly, prompt tuning can achieve and sometimes surpass the performance of direct fine-tuning when the model size is large enough. It exhibits better generalization and enhanced transferability. However, direct fine-tuning, as a more straightforward approach, is worth considering when training samples are abundant, as it allows for stable performance leveraging a smaller model, which is particularly crucial in industrial applications.

6.2 Human values detection

PLMs of medium size: From the F1 scores, it is evident that the model requires further improvement. However, the accuracy metric in the experiments reflects the model's detection capability is viable. The main direction for improvement lies in reducing false positives.

Chat-LLMs: Despite evaluating only a small subset of data, LLMs have demonstrated commendable performance in understanding human values.

6.3 Limitations

When comparing the performance of fine-tuning and prompt tuning, the inability to control model parameter sizes and trainable parameters leads to a decrease in the persuasiveness of the results.

However, such conclusions remain meaningful for practical applications because, in the face of various trade-off conditions, considerations across all aspects are often needed rather than an itemized comparison of pros and cons.

References

Georgios Balikas. 2023. John-Arthur at SemEval-2023 Task 4: Fine-Tuning Large Language Models for Arguments Classification. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval'23)*, pages 1428–1432, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Long Ma. 2023. PAI at SemEval-2023 Task 4: A General Multi-label Classification System with Class-balanced Loss Function and Ensemble Module. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval'23)*, pages 256–261, Toronto, Canada. Association for Computational Linguistics.

Nailia Mirzakhmedova, Johannes Kiesel, Milad Alshomary, Maximilian Heinrich, Nicolas Handke, Xiaoni Cai, Barriere Valentin, Doratossadat Dastgheib, Omid Ghahroodi, Mohammad Ali Sadraei, Ehsaneddin Asgari, Lea Kawaletz, Henning Wachsmuth, and Benno Stein. 2023. The touché23-valueeval dataset for identifying human values behind arguments.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Daniel Schroter, Daryna Dementieva, and Georg Groh. 2023. Adam-Smith at SemEval-2023 Task 4: Discovering Human Values in Arguments with Ensembles of Transformer-based Models. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval'23)*, pages 532–541, Toronto, Canada. Association for Computational Linguistics.

Shalom H Schwartz. 1994. Are there universal aspects in the structure and contents of human values? *Journal of social issues*, 50(4):19–45.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*. Springer.

Masaya Tsunokake, Atsuki Yamaguchi, Yuta Koreeda, Hiroaki Ozaki, and Yasuhiro Sogawa. 2023. Hitachi at SemEval-2023 Task 4: Exploring Various Task Formulations Reveals the Importance of Description Texts on Human Values. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval'23)*, pages 1723–1735, Toronto, Canada. Association for Computational Linguistics.