Fine-tuning RoBERTa on SemEval-2023 Task 4: Human Value Detection

Choi Wai Lap
The Hong Kong University of Science and Technology
wlchoiac@connect.ust.hk

Abstract

Human Values are convictions about what is good. Concretely, it is good to have our own ideas, to respect others, or to seek the truth. They are not tied to specific circumstances but are employed throughout our lives. To externalize the judgment and decision, we employ the Large Language Model (LLM) to reveal the values that we are resorted to. In this project, we fine-tune RoBERTa, a robustly optimized pre-trained bidirectional LM, to detect human values given a textual argument. We propose novel methods to optimize training and elicit model hidden representations of various human values, evaluate performance, perform ablation studies over different model candidates, and investigate the underlying reasons. Though beneficial, we show that training on a balanced dataset does not bring substantial improvement over the performance for this task. We also find out that our novel prompt-based classification objective outperforms the traditional fine-tuning methods over various training setups. Finally, we open-sourced all the codes for reproducibility[1].

1 Introduction

In today's interconnected world, understanding and respecting human values have become essential for fostering harmonious relationships and promoting inclusivity. Human values encompass a wide range of principles, such as individual autonomy, empathy, honesty, and fairness, which guide human behavior and shape societal norms. Acknowledging the significance of these values, this project aims to leverage the power of Natural Language Processing (NLP) and machine learning techniques to automatically classify textual responses based on their underlying human values.

2 Related Work

Schroter et al. (2023) presented the best-performing approach among all submissions to the SemEval 2023 Task 4 competition in identifying human values behind textual arguments. They leveraged the transformer architecture, specifically the variations of BERT, DeBERTa (He et al., 2021), and RoBERTa Large model (Liu et al., 2019) with ensemble methods for selecting the best threshold in assigning labels to each input. They show that the training step that maximizes the f1 score does not necessarily minimize the validation score, and thus ensemble on prediction averaged over two models with each obtaining optimal performance with respect to each criterion. In this paper, we also find similar behavior throughout training, but instead, pick the model whose f1-score is highest for simplicity.

Orthogonal on their works, Monazzah et al. (2023) proposed methods to augment the imbalanced dataset provided by the task organizer using the metadata of the training data. They show considerable gains in model performance trained on the augmented dataset. They employ BERT and RoBERTa for fine-tuning and achieve the rank of 14-th in the Main test set. Inspired by their work, in this project, we have put some effort into mitigating the class imbalance on the training data and obtaining a slight improvement in the model performance, but the gain is not as inspiring as that compared to using our proposed novel prompt-based fine-tuning training objective.

[1] https://drive.google.com/drive/folders/15DjN7JGAlt0lAFWt8gdjAjLCHaM5ZMGn?usp=drive_link

Conclusion	Stance	Premise	Label
We should ban human cloning	in favor of	we should ban human cloning as it will only cause huge issues when you have a bunch of the same humans running around all acting the same.	0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
We should end the use of economic sanctions	against	sometimes economic sanctions are the only thing that will get the corrupt governments to take action.	0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Table 1: Two examples of argument instance on the training set. Each instance consists of Conclusion and Premise, stance of the premise towards the conclusion (either "in favor of" or "against"), and a label list of 1s (argument resorts to value) and 0s (argument does not resort to value). The label order follows that on Figure 1.

Model Input
Premise is that We should ban human cloning. Conclusion is that we should ban human cloning as it will only cause huge issues when you have a bunch of the same humans running around all acting the same. So, is the context related to the <MASK>?
Premise is that We should end the use of economic sanctions. Conclusion is that sometimes economic sanctions are the only thing that will get the corrupt governments to take action. So, is the context related to the <MASK>?

Table 2: Our novel approach of prompt based MLM fine-tuning by concatenating premise and conclusion and inserting task specific description (red colored) into the prompt. During back-propagation, losses are calculated only on the output vector representation of the mask token (green colored).

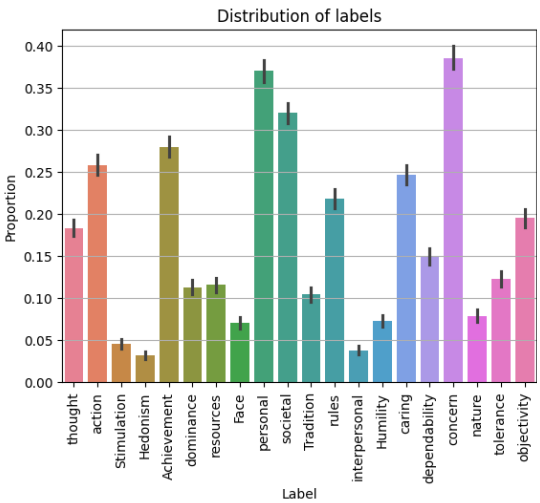

Figure 1: Labels distribution of the training set.

3 Background

3.1 Touché23-ValueEval Dataset

The Touché23-ValueEval Dataset (Mirzakhmedova et al., 2023) contains 9324 arguments from 6 diverse sources, covering religious texts, political discussions, free-text arguments, etc. Each argument was annotated by 3 crowd workers for 54 values. The dataset consists of a 2-level taxonomy of human values, with a higher level capturing the border aspect of human values behind arguments. In this project and this task, participants are asked to identify the human values on the second level, which consists of 20 categories. Note the nature of this task is a multi-label multi-class classification task since multiple values can be resorted to one text (see Table 1 for samples).

3.2 BERT Variants

RoBERTa (Liu et al., 2019), an extension of BERT, builds upon its predecessor by introducing modifications to the pre-training process. It employs a larger pre-training dataset and trains the model for a longer duration, resulting in improved performance. RoBERTa removes certain pre-training objectives like next sentence prediction, which allows the model to focus solely on the masked language modeling task. This modification leads to better language understanding and representation capabilities.

DeBERTa (He et al., 2021), another variant in the BERT family, incorporates additional improvements to enhance the model's performance. It introduces disentangled attention, which enables the model to independently attend to different aspects of the input, capture more fine-grained information and better understand the relationships between words and their contextual dependencies. By decoupling the attention heads, DeBERTa achieves superior performance compared to its predecessors in various NLP tasks.

In this project, we conduct an ablation study by varying different model architectures and sizes. By comparing their performance, we hope to identify the underlying reason for their successes and outline the key factors that contribute the most in the domain of human values classification.

3.3 In-Context Learning

The versatile in-context few-shot learning ability introduced in GPT-3 (Gao et al., 2021) allows the model to perform task-specific fine-tuning based on a few examples or demonstrations. By showcasing desired responses or actions, users can shape and customize the model's output, making it more tailored to their specific requirements. Moreover, in-context learning enhances the model's contextual understanding. By considering the context of the conversation or task at hand, GPT-3 can generate responses that are more coherent and relevant to the specific context. This contextual awareness improves the user experience and makes the model more effective in generating high-quality and meaningful responses.

Xie et al. (2022) deduce a mathematical formulation to explain in-context learning as implicit Bayesian inference, where in-context learning only occurs when a prompt can "locate" a previously learned concept through prompt distribution. Concretely, the pre-training distribution of modern LLM has already captured the affluent document-level latent information, including syntactical and symmetrical statistics. They show that if the prompt whose distribution overlaps with the pre-training distribution, in-context learning occurs and can steer the model generation toward the prompt context.

To this end, in this project, we propose to manually instantiate in-context learning by incorporating prompts whose distribution echoes with the pre-training dataset. Since the pre-training distribution of RoBERTa didn't incorporate Touché23 for sure, we follow Delvin et al. (2018) to "instill" knowledge into the distribution stored inside the model's parameters by first applying MLM-objective on the training set followed by our novel prompt-based fine-tuning (§4.2).

3.4 Prompt-based Fine-Tuning

During standard head-based fine-tuning, we typically use $\tilde{x}_{single}$ = `[CLS]` $\tilde{x}_1$ `[SEP]` or $\tilde{x}_{pair}$ = `[CLS]` $\tilde{x}_1$ `[SEP]` $\tilde{x}_2$ `[SEP]` as the input format. For downstream classification tasks with a label space $\mathcal{Y}$, we usually train a task-specific head, $\texttt{softmax}(\boldsymbol{W}_o \boldsymbol{h}_{\texttt{[CLS]}})$, by maximizing the log probability of the correct label. Here, $\boldsymbol{h}_{\texttt{[CLS]}} \in \mathbb{R}^d$ represents the hidden representation of `[CLS]`, and $\boldsymbol{W}_o \in \mathbb{R}^{|\mathcal{Y}| \times d}$ denotes a set of randomly initialized parameters introduced at the beginning of fine-tuning. However, incorporating these new parameters can be challenging when working with limited annotated data, as they can significantly increase the model's parameter count (e.g., 2,048 new parameters for a RoBERTa-large model in a simple binary classification task).

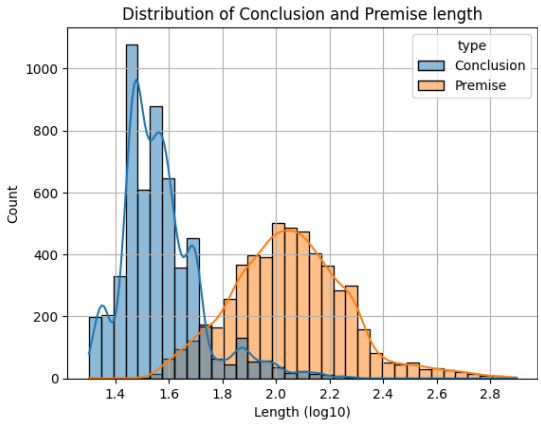

Figure 2: Input distribution of the training set.

To address this challenge, Gao et al. (2021) proposed prompt-based fine-tuning. In prompt-based fine-tuning, the model $\mathcal{L}$ is directly trained to "auto-complete" natural language prompts. For a binary sentiment classification task, they construct a prompt with input $\tilde{x}_1$ = "No reason to watch it." as $\tilde{x}_{prompt}$ = `[CLS]` $\tilde{x}_1$ It was `[MASK]`. `[SEP]`. By presenting this prompt to $\mathcal{L}$, they let the model decide whether it is more appropriate to fill in the masked token with "great" (indicating a positive sentiment) or "terrible" (indicating a negative sentiment). Note that this method does not induce any additional parameters and also matches the pre-training objective more, enabling more effective learning even with limited annotated data.

4 Implementation

In this session, we will discuss the implementation details of our methods. We first detail why (§4.1) and how (§4.2) we modify the original prompt-based fine-tuning method to suit our desired objective, followed by the (§4.3) data processing steps, (§4.4) training settings, and (§4.5) the description of our two-stage fine-tuning pipeline.

4.1 Incompatibility

The prompt-based fine-tuning method operates under two assumptions: (A) the label is binary, and (B) the tokenization method does not split label words into multiple tokens. Regarding (A), the

model only needs to make a single prediction in its input, which works well for single-label multi-class classification tasks. However, our task involves multi-label multi-class classification, where each input sentence may have multiple candidate label words. Dynamically augmenting the prompt by inserting the correct number of `[MASK]` tokens in $\tilde{x}_{prompt}$ (e.g., if $\tilde{x}_1$ is associated with 6 human values, then we put 6 `[MASK]` tokens in the prompt), is not feasible in practice, as it would provide illegal information to the model. Similarly, for assumption (B), different tokenizers tokenize human values into input IDs of varying lengths. For example, "Hedonism" may be tokenized as `['he', '##don', '##ism']` while "Tolerance" is tokenized as a whole for the BERT base tokenizer. This discrepancy in tokenized lengths for different human values within input sentences makes it challenging to provide the model with prompts in a uniform template.

4.2 Semi-Prompt-Based Fine-Tuning

To enjoy the benefit of both in-context learning and prompt-based fine-tuning, we propose a compromise approach for incorporating both methods together by framing our task as a semi-prompt-based masked language modeling (MLM) problem. The model receives an input consisting of a task-specific template that includes detailed instructions for the task, along with a masked token. Attached to the final layer of the language model is a linear head that maps the hidden dimensions of the model to the class dimensions, which is set to 20 for our project. During the loss calculation, only the hidden vector corresponding to the masked position is passed to the linear head. We employ binary cross-entropy loss as the loss function for catering to multi-label classification needs.

To this end, we can leverage the benefits of in-context learning by utilizing custom, tailor-made prompts to guide the model's generation process toward our specific classification task (see Table 2). This enables us to shape the model's responses in a way that aligns with our desired objectives. Additionally, the prompt-based fine-tuning approach ensures that the model output remains closely tied to the pre-training objective. Since BERT and its variants are primarily trained on MLM objectives, sticking to the same setting as the pre-training phase allows us to effectively unleash the model's capacity, which aligns with the idea of DeBERTa.

A notable advantage of this setup is the flexibility it offers. By tuning our prompt template multiple times, we can easily modify the training setting without the need to change any model parameters or structure. This allows us to iterate and experiment with different prompts, refining the model's behavior and improving its performance without the overhead of extensive retraining.

Our approach has similarities to the conventional classification setup, as both utilize the hidden vector corresponding to a special token (head-based: `[CLS]`; ours: `[MASK]`) and pass it through a linear head for classification. However, there is a key distinction. In our approach, we incorporate in-context learning by providing task-specific instructions within the prompt, with the `[MASK]` token serving as a shifted `[CLS]` token. The position of the `[MASK]` token becomes significant and varies across training sequences. Placing the desired `[MASK]` token at different positions has a substantial impact on the input context and model's performance, as prior research (Gao et al., 2021) has demonstrated the importance of prompt templates. Varying prompt templates can alter model performance dramatically.

4.3 Data Processing

To optimize computation and minimize memory usage, we employ dynamic padding for the input sequences in our training dataset, which varies in length (see Figure 2). By grouping sequences into batches where each batch has the maximum length of the training sample within that batch, we efficiently pad the batches on the fly. This dynamic padding approach has proven to greatly enhance training speed and reduce GPU memory consumption.

We conducted multiple ablations to explore different combinations of prefixes, suffixes, and infixes for the prompt template. Interestingly, we found that including stances, which indicate support or opposition to an argument, did not have a significant impact on the model's performance. This observation aligns with the nature of our task, which involves classifying human values. The underlying topics and values conveyed by the sentences should remain unchanged, regardless of the stance taken. Therefore, we decided to exclude stances from the prompt and only provide the model with the premise and conclusion. To guide the model's understanding of our objectives, we introduced task-specific words into the prompt. The

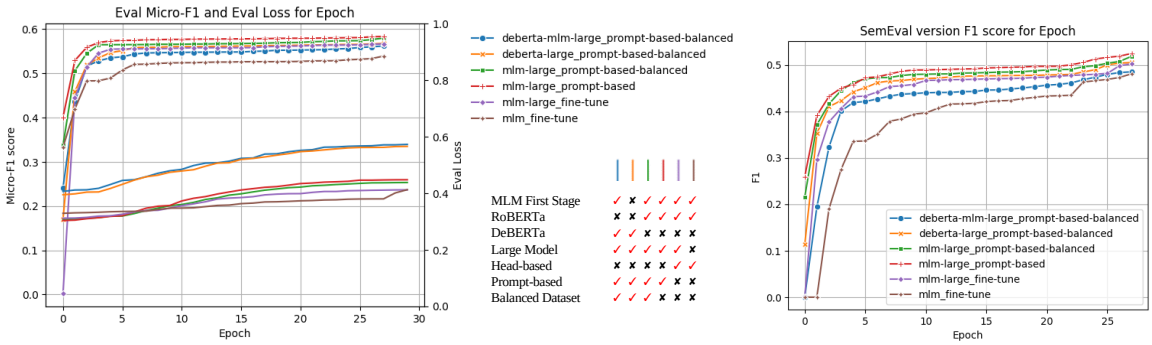

Figure 3: Model performance on the evaluation dataset over epochs. Evaluated in (left axis of left plot) Micro-F1 score and (right axis of left plot) Loss and (right plot) SemEval version F1 score (The used f1-score differs slightly from the implementation of f1-score in sklearn). Detailed discussion can be found on §5.

Model	RoBERTa					DeBERTa	
Model Size	Base		Large				
Experiment	Baseline	MLM + F.T.	MLM + F.T.	MLM-Prompt Based	↑ Balanced	Prompt Based Balanced	MLM Prompt Based Balanced
Accuracy	43	43	46	**47**	**47**	45	18
Micro F1	54	54	58	**59**	**59**	57	31
Micro Precision	60	60	63	65	**67**	62	18
Micro Recall	50	50	53	54	53	52	**89**
Thought	49	47	55	**59**	52	53	18
Action	59	61	67	**70**	68	66	41
Stimulation	9	11	9	16	10	**19**	10
Hedonism	35	22	33	**36**	27	22	12
Achievement	57	56	61	59	**63**	57	42
Dominance	27	25	32	**41**	36	**41**	22
Resources	46	40	44	**48**	44	47	10
Face	8	2	24	25	15	**29**	11
Personal	74	73	74	75	77	74	51
Societal	58	61	62	**63**	62	**63**	40
Tradition	55	48	51	54	**57**	52	17
Rules	48	46	49	49	**52**	50	31
Interpersonal	24	29	26	30	17	**43**	7
Humility	12	8	10	15	7	**17**	9
Caring	46	50	52	57	**59**	52	35
Dependability	15	24	28	20	27	**29**	20
Concern	73	71	74	75	**77**	71	54
Nature	73	72	**81**	**81**	75	80	19
Tolerance	37	31	37	**38**	36	36	22
Objectivity	41	40	**49**	43	37	42	46

Table 3: Model performance across different configurations on test set. Detailed discussion can be found in §5.

final prompt template, outlined in Table 2, incorporates these adjustments.

4.4 Training Setting

In line with Liu et al. (2019), we employ a super-low learning rate of 5e-5 with a linear learning rate scheduler and apply gradient clipping with a norm of 1 during each gradient update to ensure training stability. Additionally, we utilize a weight decay of 0.001 and implement mixed precision training, converting tensors to float16 data type for faster training. Model accuracy, recall, precision, and f1 score are evaluated for both macro and micro metrics on the evaluation set after each training epoch. We train the model for 30 epochs, using a batch size of 256 for training, evaluation, and testing to ensure sufficient information for the model's learning and efficient knowledge-updating processes.

4.5 Fine-Tuning Stage

The fine-tuning process consists of two stages. Firstly, we conduct fine-tuning on the training dataset using the masked language modeling (MLM) training objective, where each training sample is randomly masked out with a 0.2 probability. This aligns with the pre-training setting of BERT (Devlin et al., 2018). Subsequently, we select the best checkpoint, which achieves the lowest evaluation loss on the evaluation set, and proceed to the second stage. In this stage, we employ our semi-prompt-based fine-tuning method, as discussed earlier (§4.2), to adapt the model for our specific multi-label multi-class classification task. The rationale behind the first stage is to incorporate knowledge from the training distribution into the model parameters. This is crucial because in-context learning requires the prompt distribution to

correspond or align with the pre-training distribution (see §3.3). By performing MLM training, we aim to increase the likelihood of successfully triggering in-context learning in the second stage. The objective of the second stage is the primary focus of our task.

5 Ablation Study

We conduct a comprehensive performance comparison by exploring various combinations, such as different models, model sizes, balanced datasets, fine-tuning based on the masked language modeling (MLM) training objective (as referred as the first stage), and our prompted-based fine-tuning (referred as second stage). Given the extreme class imbalance in the dataset (as shown in Figure 1), we report the micro F1 score for each class to ensure fairness, as it considers the class distribution. All evaluations shown in Table 3 are performed on the testing set, with the model loaded with the best micro F1 score achieved on the evaluation set. The calculation of metrics can be performed using the SciKit Learn (Pedregosa et al., 2011) and Hugging Face (Wolf et al., 2020) libraries. In the following section, we will discuss the model's performance over configurations and provide details on the experimental settings.

For brevity, we refer to the *baseline* model used throughout the rest of this paper as "RoBERTa-base." This model is fine-tuned using the conventional head-based training objective. Specifically, we employ the "`roberta-base`" checkpoint on the `RobertaForSequenceClassification` class from the prominent `transformers` library and utilize the hidden vector representation of the `[CLS]` token to predict the desired human values across inputs, with a Binary Cross Entropy loss to facilitate the nature of multi-label classification.

It is important to highlight that the calculation of the f1-score in this competition differs slightly from our usual implementation with the scikit-learn library. In this competition, they utilized a specific formula for calculating the f1-score:

$$f1 = \frac{2 \times recall_{macro} \times precision_{macro}}{recall_{macro} + precision_{macro}}$$

This approach differs from the usual method of calculating the f1-score for each label and then taking the average. As a result, we provide scores for both versions of the calculation in Figure 3 to provide a comprehensive comparison.

The left plot in Figure 3 represents the micro f1-score derived from the scikit-learn library, which follows the usual implementation. On the other hand, the right plot in Figure 3 represents the SemEval version of the f1-score. Both plots showcase the results obtained on the evaluation dataset.

5.1 Baseline vs MLM F.T.

Based on the results presented in Table 3, we compare the model performance between the baseline configuration and the MLM F.T. (Fine-Tuning) configuration. Both configurations utilize the same base model (`roberta-base`), with the only difference being that the MLM F.T. configuration undergoes pre-training on the MLM objective (first stage) before fine-tuning on the conventional head-based classification task. Interestingly, we did not observe any significant difference in terms of model performance between these two configurations. They achieve similar F1 scores and accuracy in each class. This suggests that injecting the training distribution onto the model parameters or knowledge before fine-tuning is not effective for the conventional head-based objective method. This outcome is expected because the purpose of the first stage is to align the pre-training distribution with the prompt distribution to facilitate in-context learning. However, since the MLM F.T. configuration directly feeds the model with the premise and conclusion in the input without any prompt template, the model may struggle to identify the prompt distribution without additional context, leading to a similar performance as the baseline configuration.

5.2 MLM F.T. vs MLM Prompt-Based

Next, we proceed to compare the performance of the MLM F.T. (Fine-Tuning) configuration with the MLM Prompt-Based configuration. In the latter configuration, the task-specific prompt description (see Table 2) is utilized in the prompt rather than directly feeding the model with raw input data. Both configurations are trained using the large version of RoBERTa and undergo the first stage MLM objective before fine-tuning in the second stage.

Upon examining Table 3, we can observe that the prompt-based fine-tuning objective outperforms the conventional head-based objective. The testing F1 scores for the prompt-based configuration are consistently higher on average compared

to the scores for the head-based configuration. This indicates that in-context learning is indeed triggered and contributes to improving the model's understanding of our desired task.

5.3 Balanced Dataset

Considering that it is a common practice and an industry standard to balance datasets before fine-tuning, we proceed to compare the model performance between the MLM Prompt-Based configuration and the same configuration but applied to a balanced dataset. However, due to the multi-label nature of the dataset, it is challenging to achieve perfect label balance across every class. Instead, we modify the dataloader sampling strategy by explicitly defining class weights. This favors the sampling of training data associated with less frequent class labels in the overall label distribution. Additionally, we assign higher weights to minority classes proportional to their label population during the loss calculation. Both weights can be obtained by `compute_class_weight` from scikit-learn.

Both configurations achieve similar performance on average, as shown in Table 3. Furthermore, the evaluation loss and micro F1 curves over epochs (depicted by the red and green lines in Figure 3) exhibit the same trajectory for both configurations. This indicates that the choice of a balanced dataset does not significantly improve the model's performance for this specific task. We hypothesize that since the dataset is relatively small (9K samples) and exhibits a skewed class distribution (see Figure 1), balancing the data has limited effectiveness, resulting in similar performance between the two configurations.

5.4 DeBERTa

Finally, we proceed to compare the performance between the prompt-based configuration and the MLM prompt-based configuration on the DeBERTa large model. These two configurations differ in whether or not they undergo the first stage (MLM) before fine-tuning in the second stage. We were intrigued by the success of the champion team (Schroter et al., 2023) in this SemEval Task 4 competition, who utilized the state-of-the-art DeBERTa model for ensemble learning. We aimed to investigate whether DeBERTa could provide insights into our desired task.

However, upon examining Table 3, we did not observe any significant improvement in performance for both configurations compared to the RoBERTa MLM prompt-based configuration. In fact, the average F1 score of DeBERTa was slightly worse than that of RoBERTa. We posit that our prompt-based fine-tuning strategy does not naturally align with the capabilities of the DeBERTa model. Interestingly, the champion team's implementation followed the standard head-based fine-tuning method on the hidden representation of the `[CLS]` token in the DeBERTa model for ensemble learning, without any modifications to the prompt. We hypothesize that further improvements could be achieved by adopting their approach and incorporating an ensemble strategy to determine the probability threshold for assigning labels.

However, since this project is primarily experimental and aimed at exploring our novel prompt-based fine-tuning strategy, we leave the exploration of ensemble methods and probability threshold assignment as future work.

6 Conclusion

In this project, our focus was on fine-tuning the RoBERTa model and its variants for the SemEval Task 4: Human Value Detection task. To enhance the performance of the models, we introduced a novel prompt-based fine-tuning strategy and incorporated MLM objectives. The aim was to align the pre-training distribution with the prompt distribution, thereby facilitating in-context learning and guiding the model's generation process.

We conducted extensive ablation studies, exploring various configurations such as different model architectures, sizes, training objectives, and dataset balancing techniques. Through these studies, we aimed to identify the most effective approaches for improving model performance.

Ultimately, based on the evaluation dataset, we selected the best checkpoint from our top-performing model, which was RoBERTa-Large with MLM as the first stage and prompt-based fine-tuning as the second stage. This model achieved a SemEval version of the f1-score of 0.5 on the test set. As a result, (if we participated in the competition) we would be ranked 8th out of 41 teams, surpassing their baseline model (BERT) provided by the task organizers which ranked 30th with an achieved f1-score of 0.42.

References

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. arXiv preprint arXiv:2012.15723v2.

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. arXiv preprint arXiv: 1907.11692.

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. arXiv preprint arXiv: 2301.13771.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Sang Michael Xie, Aditi Raghunathan, Percy Liang and Tengyu Ma. 2021. An Explanation of In-context Learning as Implicit Bayesian Inference. arXiv preprint arXiv: 2111.02080.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Pedregosa, Fabian, Fabian Pedregosa@inria, Fr, Gael Org, Vincent Michel, Bertrand Fr, Olivier Grisel, et al. "Scikit-Learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT et AL. Matthieu Perrot Edouard Duchesnay." Journal of Machine Learning Research 12 (2011): 2825–30. https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Alexander M. Rush. (2020) HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv preprint arXiv: 1910.03771.

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. arXiv preprint arXiv: 2305.08625v1.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv preprint arXiv: 2006.03654.

Erfan Moosavi Monazzah, and Sauleh Eetemadi. 2023. Prodicus at SemEval-2023 Task 4: Enhancing Human Value Detection with Data Augmentation and Fine-Tuned Language Models. Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 2033–2038 July 13-14, 2023 ©2023 Association for Computational Linguistics.