Exploring LLM Priming Strategies for Few-Shot Stance Classification

Yamen Ajjour and Henning Wachsmuth
Leibniz University Hannover, Institute of Artificial Intelligence
<initial>.<lastname>@ai.uni-hannover.de

Abstract

Large language models (LLMs) are effective in predicting the labels of unseen target instances if instructed for the task and training instances via the prompt. LLMs generate a text with higher probability if the prompt contains text with similar characteristics, a phenomenon, called priming, that especially affects argumentation. An open question in NLP is how to systematically exploit priming to choose a set of instances suitable for a given task. For stance classification, LLMs may be primed with few-shot instances prior to identifying whether a given argument is *pro* or *con* a topic. In this paper, we explore two priming strategies for few-shot stance classification: one takes those instances that are most semantically similar, and the other chooses those that are most stance-similar. Experiments on three common stance datasets suggest that priming an LLM with stance-similar instances is particularly effective in few-shot stance classification compared to baseline strategies, and behaves largely consistently across different LLM variants.

1 Introduction

Large language models (LLMs) have enabled a new input paradigm in NLP by following instructions that define the task to be solved: prompting. Designing optimal instructions for a given task is a key challenge in this paradigm. A common technique in prompt engineering is to append a set of few-shot instances to the instructions that are similar to the target instance. Although this technique is widely used, research lacks a clear understanding of what makes a set of examples effective for a target instance (Min et al., 2022). A mechanism that helps to explain the effect of a prompt on the output of an LLM is priming, which is the effect of a certain stimulus (prime) on processing a subsequent stimulus (target) (Misra et al., 2020).

Priming influences human behavior by making certain information more salient and accessible. Research on argumentation in political science studies how priming connects media exposure with voting behavior. Media coverage highlights a candidate's topics and arguments to increase the chances that voters elect the candidate (DellaVigna and Kaplan, 2007; Iyengar and Hahn, 2009). The way the topics are presented with either positive or negative sentiment in news influences how the audience votes, which is called affective priming (Kuehne et al., 2011). For LLMs, appending a set of similar instances to the target instance in a prompt can be seen as priming for label voting.

Prompting research has shown that the choice and order of training instances have a strong effect on model performance. Among others, Liu et al. (2022) find that semantically similar instances are most effective in sentiment analysis, question answering, and text-to-table generation. However, it is unclear so far whether this finding generalizes to tasks dealing with argumentation, such as stance classification: classifying an argument as *pro* or *con* towards a controversial topic (Somasundaran and Wiebe, 2009).

In this paper, we study how to choose the best training instances for few-shot priming in stance classification. We investigate two alternative priming strategies: prompting an LLM with training instances that are (a) semantically similar to the instance to be classified or (b) stance-similar (e.g., pro electric cars and con fuel cars). While the first builds on the idea of Liu et al. (2022) and semantic priming, the second builds on affective priming. We contrast both priming strategies to diversification, which has been observed to foster better performance in stance classification (Schiller et al., 2024; Arakelyan et al., 2023).

To operationalize the priming strategies, we use contrastive learning to quantify the similarity between training instances and a given target instance. The first strategy, `semantic-priming`, returns the k instances with the highest semantic similarity.

The second, affective-priming, returns k instances with the highest stance similarity. Finally, the diversification strategy, distinct-k, groups the training instances into k clusters according to their semantic similarity and uses the most central representative of each cluster as a prime. Figure 1 contrasts the three priming strategies.

We evaluate all priming strategies against random sampling on three widely used stance classification datasets, IBMSC (Bar-Haim et al., 2017), VAST (Allaway and McKeown, 2020), and Perspectrum (Chen et al., 2019). We employ four different LLMs in two manners: Llama2-7b (Zhang et al., 2022) and Vicuna-7b (Chiang et al., 2023) in prompting, as well as Alpaca-7b (Taori et al., 2023) and Mistral-7b-instruct (Jiang et al., 2023) in both prompting and instruction fine-tuning. According to our results, affective-priming shows substantial improvements over random sampling and diversification in prompting for Llama2-7b and Vicuna-7b. semantic-priming is more effective when the number of shots is low (up to 4).

Our findings contribute to research in three ways: (1) We investigate for the first time the effect of affective priming on large language models. (2) We establish priming strategies as a central component of approaches to few-shot stance classification. (3) We advance the state-of-the-art on stance classification on IBMSC and Perspectrum.[1]

2 Related Work

Prompting defines a task as instructions that an LLM completes with the desired output. Few-shots are exemplary instances of the task together with their expected outputs that are added to the instructions. The selection of few-shots is decisive for the performance of an LLM on the task. Gao et al. (2021) show that prepending the input instance with semantically similar instances to it is more effective in four GLUE tasks (Wang et al., 2019) than using random instances. Like us, they use SBERT (Reimers and Gurevych, 2020) to encode the instance to be classified and the few-shot instances, but they do not investigate what similarity is effective for a given task.

Liu et al. (2022) find that GPT-3 exploits similar instances more than random ones, improving effectiveness on sentiment analysis and table-to-text generation. Levy et al. (2023) use BM25 similarity to sample diverse instances for semantic parsing, outperforming a sampling of similar instances. We consider prepending instances that are similar to the input instance to the instructions as priming. Instead of using vanilla similarity measures, we propose a contrastive-learning-based similarity measure to retrieve few-shot instances that are motivated by priming theory.

Research on priming first investigated how exposure to certain stimuli influences subsequent behavior or cognition. Earlier studies show that people more effectively recognize a string as a word after being exposed to semantically similar ones (Meyer and Schvaneveldt, 1971), known as semantic priming. In political discourse, the focused coverage of topics associated with a candidate in the news makes voters more likely to vote for them in elections. In contrast, affective priming utilizes the (positive or negative) tone in which messages are conveyed to shape the attitude towards a topic (Sheafer, 2007; Kuehne et al., 2011). Following these ideas, we contrast two priming strategies that exploit semantic and stance similarity, respectively, between the training and the target instances.

Studies show that the text generated by LLMs can also be steered by priming. Misra et al. (2020) find evidence that BERT is more likely to correctly predict a masked target word in a sentence once the sentence is prepended with a semantically similar prime. LLMs also adapt to the structure of the prompt and generate text with similar syntax of an input prime (Prasad et al., 2019; Jumelet et al., 2024). While LLMs have been shown to be steered by semantic and syntactic priming, their sensitivity to positive and negative sentiment (affective priming) has not yet been explored.

Stance classification is the task of identifying the polarity of an argument towards a topic among a set of labels, such as *pro* or *con* (Somasundaran and Wiebe, 2009; Reuver et al., 2024). Researchers propose approaches that integrate the context of the target instance by learning topic representations (Augenstein et al., 2016; Wei and Mao, 2019) or retrieving related knowledge to the instance from a knowledge graph (Liu et al., 2021). In contrast to these approaches, our work shows that training instances with similar stances are helpful for prompt-based stance classification.

Few-shot stance classification aims at settings where only few training data is available (Allaway and McKeown, 2020). Prompt-based approaches either inject topic knowledge (Beck et al., 2023)

[1] Code available here: https://github.com/webis-de/priming-strategies-for-stance-classification

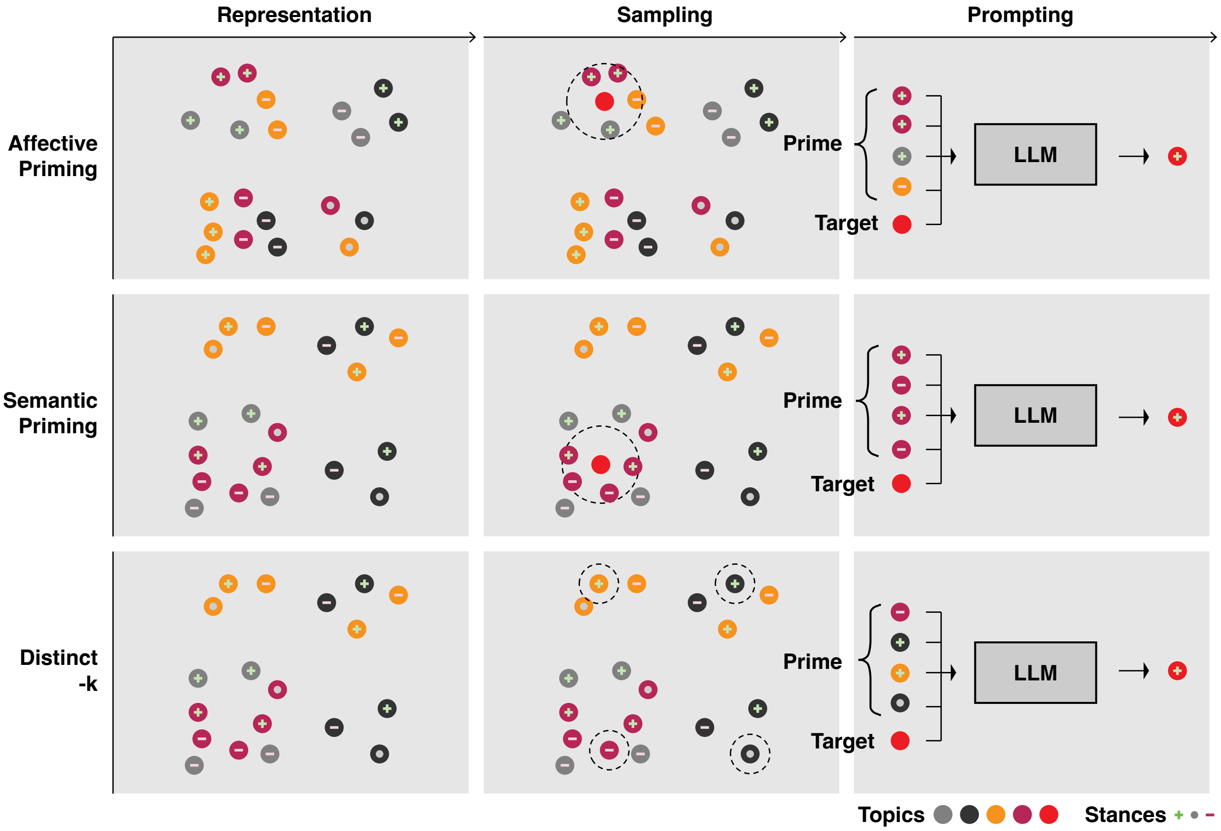

Figure 1: Comparison of the three priming strategies (`affective-priming`, `semantic-priming`, and `distinct-k`). The *representation* focuses either on stance or semantics. Sampling picks few-shots either by similarity or by diversity. Prompting combines the (here, four) few-shots with the target instance (shown in red) to classify.

or use a stance label representation (Jiang et al., 2022) in the prompt. Research on few-shots in stance classification is limited to selecting diverse instances. Arakelyan et al. (2023) proposed a diversification approach that outperforms the state of the art on several stance classification datasets. Schiller et al. (2024) analyze the effect of increasing the count of topics in the training set against increasing the size of samples per topic. Their experiments illustrate that, for small LLMs such as Ernie 2.0 (Sun et al., 2020), diversifying the training set in terms of topics improves performance on unseen topics. In contrast to diversification-based approaches, our study suggests that stance-similar instances are most effective for prompting certain LLMs (e.g., Vicuna-7b).

3 Approach

As discussed in the introduction, priming utilizes existing associations between a pair of concepts, called the *prime* and the *target*. It rests on invoking an effect on the target by mentioning the prime. Our priming approach to stance classification treats a test instance as a target and retrieves k semantically similar or stance-similar instances as training instances. The approach employs prompt-based learning to prime an LLM with the retrieved k few-shots to predict the stance of the target. In the following, we start by describing our prompt-based learning methods, which we employ for stance classification. Then, we present our priming strategies.

3.1 Prompt-based Learning

We adopt two prompt-based learning methods for language models: *prompting* and *instruction fine-tuning*. Both methods use $k \geq 1$ training instances in a few-shot manner. Each instance contains a topic, an argument, and a stance. We rely on greedy decoding in both methods to let an LLM complete the prompt with the most probable token, which is the stance label. We use the following prompt to describe the stance classification task (see Table 2 in the Appendix for the prompt template):

> “Classify the stance of the following argument on the given topic into: Pro or Con.”[2]

[2] For VAST, we add the label Neutral.

Prompting Here, we simply append the learning instances to the prompt without any fine-tuning. We use this method for four large language models (LLMs): *Mistral-7b-instruct* (Jiang et al., 2023), *Alpaca-7b* (Taori et al., 2023), *Vicuna-7b* (Chiang et al., 2023), and *Llama2-7b* (Touvron et al., 2023).

See Table 2 again for the format of the training instances. In case the training instances exceed the allowed input length of an LLM, we cut the last part of each training instance.

Instruction Fine-tuning While prompting is efficient and easy to employ since no training is required, instruction fine-tuning pushes the use of the prompt further in that the language model is fine-tuned on instruction data. Following this method, we fine-tune Alpaca-7b (Taori et al., 2023) and Mistral-instruct-7b (Jiang et al., 2023) using LoRa (Hu et al., 2022) on the k instances with an instruction prompt. The topic and argument are then given in the input section of the prompt. For fine-tuning both models, we used grid-search to find the best hyperparameters on the validation sets of the respective dataset, which we will introduce in Section 4. Full hyperparameters of both models can be found in Table 7 in the Appendix. We fine-tune the models in two steps. First, we fine-tune the models on all the training data of each dataset using the aforementioned prompt without few-shots. Second, we fine-tune the models with the aforementioned prompt on the few-shots sampled by the priming strategies from the training set.

3.2 Priming Strategies

In the following, we introduce two priming strategies that exploit stance similarity and semantic similarity between a target instance and the training instances. Afterward, we describe baseline priming strategies that are tailored to contrast the priming strategies and to analyze the strengths and weaknesses of all strategies: `distinct-k` and `random`. Our hypothesis is that training instances that are similar to the target instance in terms of semantics or stance are more effective than diverse or random training instances. Figure 1 illustrates how each of the three approaches represents, samples, and prompts instances.

Affective priming Prompting an LLM with arguments that hold similar stances to the target instance provides the most consistent stimulus to it, inducing bias in line with the original idea of priming. To this end, we train a contrastive learning embedding that captures the *stance similarity* between the instances on the training set. For training this embedding, we use SBERT (Reimers and Gurevych, 2020) and use argument pairs on the same topic with the same stance as positive instances. Argument pairs on the same topic with different stances are provided as negative instances.

For each instance, we concatenate the topic and argument, separated by `[SEP]`. Among the possible models for SBERT[3], we use the standard model `all-mpnet-base-v2`. The priming strategy then returns the k most stance-similar training instances to a given test instance in terms of cosine similarity. We make sure that this priming strategy retrieves one instance per topic to maximize the learning effect.

Semantic Priming This priming strategy assumes that the instances most semantically similar to a test instance should be chosen to prime the LLM. Accordingly, we retrieve the most *semantically* similar training instances for each test instance. The similarity is calculated by embedding a pair of training and test instances using the original SBERT embeddings and calculating their cosine similarity. Similar to our affective priming strategy, we use the standard model `all-mpnet-base-v2` among the available models for SBERT. In contrast to `affective-priming`, we select semantically similar instances while maintaining a balanced stance distribution of the selected set.

Distinct-k This baseline priming strategy assumes that a diverse selection of instances should be chosen to prime the LLM. The rationale behind this strategy is that since the training set is limited in size, it might not contain similar instances for some target instances. Following this idea, we cluster the instances in the training set into k clusters. Then, we take the top 10 nearest arguments to each cluster centroid as candidates according to Euclidean distance.[4] This allows us to ensure a balanced stance distribution in the chosen instances. To cluster the arguments, we first embed them with SBERT and then apply agglomerative clustering with Ward linkage and Euclidean distance. During training, we sample one instance from each of the cluster candidates.

[3] SBERT, `https://www.sbert.net/`

[4] For VAST, we took the top 50 instances since the class distribution in VAST is skewed (See Table 3).

Random To assess the impact of priming, we compare all strategies to `random` sampling, which takes a different random sample of size k from the training as few-shots for each test instance.

4 Experiments

The proposed priming strategies stimulate large language models to tackle stance classification using semantic and stance similarity. In the following experiments, we compare the priming strategies on different stance classification datasets.

4.1 Data

For evaluation, we require data with sufficient and representative coverage of topics to assess the robustness of our approach on unseen topics. Hence, we choose the following datasets:

IBMSC This dataset contains 2,394 arguments that are labeled as *pro* or *con* with respect to 55 controversial topics (Bar-Haim et al., 2017). The dataset is split into a training set and a test set that covers 25 topics and 30 topics, respectively. The distribution of the stance labels in the test set is almost balanced, with 48% of the arguments being con and 52% arguments being pro.

VAST This dataset contains 15,956 comments labeled as *pro*, *con*, or *neutral* with respect to 5,630 topics (Allaway and McKeown, 2020). We choose the VAST zero-shot setting, which ensures a disjoint topic selection between the training and test sets.

Perspectrum This dataset contains 11,822 claims on 907 topics that have been posted on the debate portal *debate.org* (Chen et al., 2019). Similar to IBMSC, the claims are labeled with *pro* or *con* with respect to the topic, and mostly have a balanced distribution. Details of the splits for the three datasets can be found in Table 3.

4.2 Baselines

To contrast few-shot prompting and instruction fine-tuning with standard fine-tuning, we further compare to the *majority* class found in the training set, and we fine-tune *DeBERTa* (He et al., 2020) on the training set to predict the stance of the argument. For the latter, we concatenate the argument and the topic and provide them as input for training (hyperparameters can be found in Table 4 in the Appendix). Moreover, we report the performance of several state-of-the-art approaches from related work on the datasets as available (Allaway and McKeown, 2020; Barrow et al., 2021; Arakelyan et al., 2023; Hanley and Durumeric, 2023; Zhang et al., 2025).

Finally, to contrast the few-shot approaches, we fine-tune Alpaca-7b and Mistral-7b-instruct on *all training* data. We combine all four models considered for *prompting* and the two models considered for *instruction fine-tuning* with all four prompting strategies (`random`, `distinct-k`, `semantic-priming`, and `affective-priming`). We compare the `affective-priming` strategy against a baseline (`Stance-similarity`) that uses the majority label of the k most similar training instances to the target instance as returned by `affective-priming`. We take 16 instances for IBMSC and Perspectrum, and 12 for VAST.[5]

4.3 Results

Table 1 lists the results of the experiment for the prompting and instruction fine-tuning approaches. The performance in all experiments is averaged over five seeds (including the follow-up analyses discussed below). At the bottom of the table is the performance of Alpaca-7b and Mistral-7b-instruct after fine-tuning them on the training set.

The results show that fine-tuning Mistral-7b-instruct on all training data yields the best classification performance, outperforming other models on this task. This shows the substantial impact of instruction fine-tuning on stance classification. In most cases, the priming strategies show consistent enhancement over the baseline priming strategies in prompting, which we discuss first.

The `affective-priming` strategy outperforms other priming strategies across all models on IBMSC and VAST, except for Mistral-7b-instruct. The performance of `affective-priming` is also higher than that of `Stance-similarity` in all cases (except Mistral-7b-instruct and Vicuna-7b on VAST). This indicates the advantages of using stance-similar instances to prime LLMs compared to relying solely on contrastive-learning similarity measures in few-shot classification. The performance of Mistral-7b-instruct is higher when prompted with diverse instances. In contrast, a consistently substantial improvement can be observed on VAST, where Llama2-7b and Alpaca-7b outperform the random priming strategy with 0.261

[5] Notice that we use for VAST multiples of 3, since it is annotated with three labels, which allows us to maintain a balanced stance distribution.

			IBMSC			VAST				Perspectrum		
	Approach	**Strategy**	**Pro**	**Con**	F_1	**Pro**	**Con**	**Neu**	F_1	**Pro**	**Con**	F_1
Fine-tuning	Majority		.681	0	.341	0	0	.525	.175	.693	0	.350
	DeBERTa		.717	.681	.699	.665	.679	.903	.749	.830	.809	.819
	Allaway and McKeown (2020)		–	–	–	–	–	–	.670	–	–	–
	Barrow et al. (2021)		–	–	.834	–	–	–	–	–	–	–
	Arakelyan et al. (2023)		–	–	.862	–	–	–	.543	–	–	.789
	Hanley and Durumeric (2023)		–	–	–	.695	.711	.905	.771	–	–	–
	Zhang et al. (2025)		–	–	–	.770	.794	–	.825	–	–	–
Contrastive Learning	`Stance-similarity`		.617	.569	.593	.544	.505	.879	.643	.770	.744	.757
Prompting	Llama2-7b	`random`	.741	.670	.705	.514	.403	.301	.409	.733	.746	.740
		`distinct-k`	.728	.681	.705	.499	.415	.259	.391	.755	.758	.756
		`semantic-priming`	.763	.535	.649	.500	.404	.173	.359	.783	.732	.757
		`affective-priming`	.704	.770	**.737***	.649	.539	.852	**.670***	.784	.774	**.779***
	Alpaca-7b	`random`	.686	.768	.727	.538	.542	.128	.393	.752	.799	.775
		`distinct-k`	.701	.744	.722	.547	.552	.055	.385	.758	.800	.780
		`semantic-priming`	.732	.733	.733	.532	.537	.122	.397	.800	.815	**.808***
		`affective-priming`	.737	.739	**.738**	.612	.681	.836	**.710***	.749	.797	.770
	Mistral-7b-instruct	`random`	.805	.837	.821	.556	.537	.605	.566	.826	.836	.831
		`distinct-k`	.863	.871	**.867***	.563	.553	.615	**.577**	.849	.840	**.845**
		`semantic-priming`	.856	.857	.857*	.514	.522	.465	.501	.839	.833	.836
		`affective-priming`	.858	.866	.862*	.529	.54	.639	.570	.844	.841	.843
	Vicuna-7b	`random`	.788	.762	.775	.545	.483	.329	.453	.812	.807	.809
		`distinct-k`	.813	.746	.779	.536	.477	.389	.467	.818	.808	.813
		`semantic-priming`	.803	.692	.747	.537	.498	.275	.437	.807	.774	.790
		`affective-priming`	.833	.811	**.822***	.560	.565	.564	**.563***	.818	.811	**.815**
Instruction fine-tuning	Alpaca-7b	`random`	.820	.801	.810	.581	.599	.779	.653	.842	.856	.849
		`distinct-k`	.807	.817	.812	.481	.648	.785	.638	.829	.849	.839
		`semantic-priming`	.824	.810	**.817**	.487	.640	.771	.633	.848	.860	**.854**
		`affective-priming`	.758	.701	.730	.529	.676	.820	**.675**	.853	.830	.842
	Mistral-7b-instruct	`random`	.920	.908	.914	.539	.558	.642	.641	.896	.882	.889
		`distinct-k`	.901	.940	.902	.568	.630	.823	.674	.913	.902	**.908**
		`semantic-priming`	.928	.920	**.924**	.579	.603	.798	.660	.907	.896	.902
		`affective-priming`	.912	.910	.911	.638	.655	.885	**.726**	.890	.904	.897
	Alpaca-7b (all training)		.796	.817	.806	.569	.636	.790	.665	.830	.849	.839
	Mistral-7b-instruct (all training)		.911	.927	.919	.643	.685	.835	.720	.941	.932	.936

Table 1: Accuracy and macro F_1-score of our prompting and instruction fine-tuned approaches with each priming strategy on IBMSC, Perspectrum, and VAST in comparison to the fine-tuned approaches. "–" indicates that the corresponding entry is not reported. Bold values indicate the best effectiveness in the few-shot settings. Significant enhancements relative to random sampling with a p-value less or equal to 0.01 are denoted by an asterisk (*).

and 0.317, respectively. This substantial improvement raises the question of what type of priming instances are actually chosen. Therefore, we analyzed the instances in the test set of VAST that are labeled correctly with Alpaca-7b when combined with `affective-priming` and wrongly when combined with the other priming strategies. We observe that about 91% of these instances are neutral instances for which the `affective-priming` strategy selected 97% neutral priming training instances. This suggests the substantial impact of consistency between the stance of the training instances and the test instance in prompting.

On Perspectrum, `affective-priming` yields the best performance across the priming strategies for Llama2-7b and Vicuna-7b. As with the other datasets, Mistral-7b-instruct is most effective when combined with `distinct-k`, slightly beating `affective-priming` (0.002 higher). However, `affective-priming` outperforms both `semantic-priming` and `random` sampling.

For instruction fine-tuning, we can observe that `semantic-priming` is the most effective among the priming strategies on IBMSC. Using this

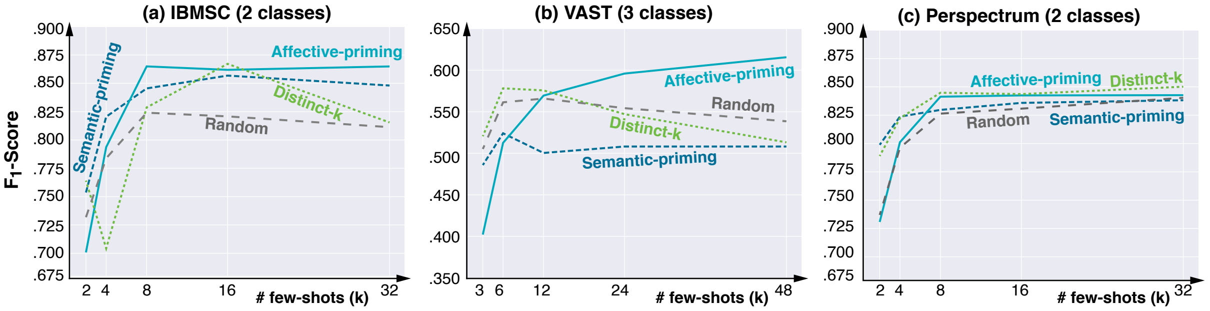

Figure 2: Macro F$_1$-score of Mistral-7b-instruct per priming strategy (`semantic-priming`, `affective-priming`, `distinct-k`, and `random`) over the few-shots k for the three datasets: (a) IBMSC, (b) VAST, and (c) Perspectrum.

strategy with Alpaca-7b and Mistral-7b-instruct yields better performance than fine-tuning them on the training set of IBMSC. On the other hand, `affective-priming` outperforms `random` for Alpaca-7b and Mistral-7b-instruct on VAST with an enhancement of 0.022 and 0.085, respectively. The performance of the `semantic-priming` strategy exceeds the performance of random on Perspectrum with a difference of 0.005 to 0.013 for Alpaca-7b and Mistral-7b, respectively. By comparing these results to those of the prompting method, we observe that priming instances are less effective than training instances in the standard instruction fine-tuning learning method. Hence, we can conclude that our priming strategies are effective when applied to prompting approaches.

5 Analysis

To further understand the priming strategies, we analyze the performance of the priming strategies for the most effective model in prompting, that is, Mistral-7b-instruct. Figure 2 shows its performance with the four strategies for a range of k values on the three datasets. As seen, `affective-priming` converges to higher performance at $k = 8$ few-shots for IBMSC and Perspectrum and at $k = 12$ for VAST. This might indicate that `affective-priming` is most effective when the stance of the test instance is repeated and consistent in the training instances. In contrast, `semantic-priming` outperforms `affective-priming` on all datasets for $k \in \{2, 3, 4, 6\}$ and saturates afterward, suggesting that, for few instances, semantic associations between the training and test instances are more effective.

Our experiments indicate that the priming strategies consistently enhance the performance of prompting methods on IBMSC and VAST. Still, they perform moderately on Perspectrum compared to diversification and `random` on all models except Llama2-7b. We can observe that `affective-priming` results in significantly better performance on IBMSC and VAST. On Perspectrum, however, the performance of `affective-priming` varies across models and is even subpar to `random` for Alpaca-7b. This raises the question of which properties of Perspectrum result in this varied performance and to which extent the priming effect is observable on this dataset.

As a first inquiry, we investigated the distribution of the similarities between the instances and target instances sampled with `affective-priming` in Perspectrum for $k = 16$. We observed that the sampled priming instances are very similar to the target instances, with a minimum value of 0.87, a mean of 0.99, and a maximum of 1. In comparison, the distribution of the similarity distribution for VAST has a minimum value of 0.44, a mean value of 0.79, and a maximum of 0.98.[6]

Since sampling instances with lower similarity results in better performance on VAST, we investigate whether sampling with lower similarities might result in better performance on Perspectrum. For this goal, we rerun the prompting experiments on Perspectrum while limiting the similarity between the prime and the target instance with a maximum threshold for `affective-priming` and `semantic-priming`. We choose thresholds that constitute increasing 10% percentiles of the similarity distribution for `affective-priming` and `semantic-priming`.

Figure 3 shows the performance of the four models in terms of macro F$_1$-score after limiting the similarity to the selected percentiles. For example, a percentile with a value of 90 means that only the

[6] The distribution of the similarity distribution for IBMSC has a minimum value of 0.40, a mean value of 0.81, and a maximum of 0.99.

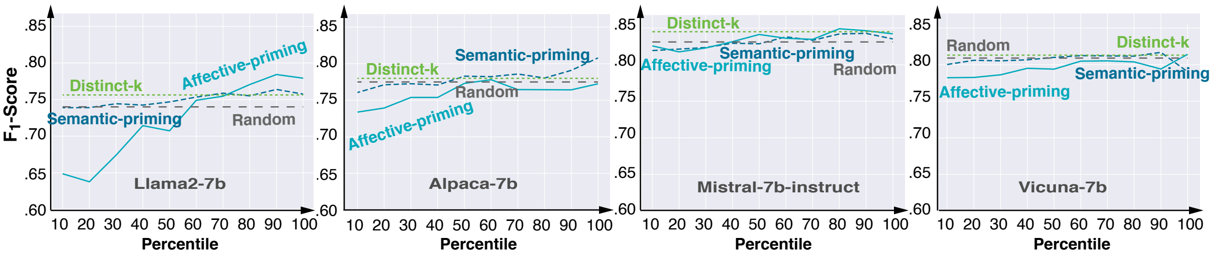

Figure 3: Macro F_1-score of Llama2-7b, Alpaca-7b, Mistral-7b-instruct, and Vicuna-7b on Perspectrum with limiting the sampled instances with increasing percentile thresholds. A percentile indicates a maximum similarity threshold applied only to affective-priming and semantic-priming. A percentile of 90 means that we only sample the 90% least similar with the priming strategies.

training instances whose similarity score to the target instances among the 90% least similar can be selected to prime the model. We also plot the performance of distinct-k and random to provide a basis to compare the priming strategy.

The figure shows that the higher the stance similarity of the prime to the target, the better the performance of the models. However, this increasing performance depends largely on the model. For example, by taking the first 10% instances to prompt Llama2-7b with affective-priming, which are the least stance-stance similar, we achieve an F_1-score of 0.648. In comparison, Llama2-7b achieves an F_1-score of 0.784 when sampling from the 90% least stance-similar instances. A similar but less steep increase can be observed for semantic-priming where Llama2-7b achieves an F_1-score of 0.739 at the percentile 10 and an F_1-score of 0.764 at the percentile 90. We also observe a small drop (around 0.005 points) in the performance for both priming strategies from the percentile 90 to the percentile 100. This might indicate that instances that are very similar to the target instances are not the best for priming the model.

According to this analysis, the effect of affective-priming on Vicuna-7b and Mistral-7b-instruct is lower than Llama2-7b but is still a substantial increase. For example, the performance of Vicuna-7b increases from 0.783 at the percentile 10 to an F_1-score of 0.815 when considering all the training instances (percentile 100). Both Vicuna-7b and Mistral-7b-instruct show a drop in performance at higher percentiles (the percentile 90 for Vicuna-7b and 100 for Mistral-7b-instruct).

This analysis corroborates the observation that certain large language models can be steered by affective-priming. It also shows that the effect of affective-priming largely depends on the model. We observe that taking highly stance-similar instances to the target instances results in some cases in subpar performance on Perspectrum. This might explain the moderate performance of the priming strategies on Perspectrum compared to IBMSC and VAST.

6 Discussion

This section discusses possible reasons for the varied performance of affective-priming across models and gives practical recommendations for selecting few-shots for stance classification.

Among the four models, our experiments demonstrate that Llama2-7b and Vicuna-7b are most susceptible to affective-priming across datasets in the prompting setup. Vicuna-7b is fine-tuned from Llama2-7b on ChatGPT conversations. Since both models are susceptible to affective-priming, the datasets on which Llama2-7b was pre-trained might be one cause for the models' susceptibility to affective priming. Datasets that contain opinionated information, such as news or online forums, might include certain associations that are triggered by the few-shots in the prompt.

Another possible reason for the difference in performance of affective priming across the models is the models' architecture. Possible design choices that can affect the sensitivity to affective priming are the attention mechanism or the activation function. Whilst our experiments are comprehensive in terms of the studied model architectures, a systematic study of the effect of the model elements on the sensitivity to priming is beyond the scope of this paper.

Finally, the model developer's application of alignment methods such as Reinforcement Learning from Human Feedback (RLHF) or other fine-tuning steps might make the model more or less susceptible to priming. While none of the four models are aligned using RLHF, all models except

Llama2-7b are instruction fine-tuned. The data or method used for fine-tuning the three instruction fine-tuned models might be one source for the varied performance of the models.

Practical Recommendations Our experiments illustrate the merit of selecting instances that are stance-similar to the input instance for few-shot stance classification. In addition, our experiments demonstrate the benefit of diversifying the training instances in terms of topic, which resonates with the work of Arakelyan et al. (2023); Schiller et al. (2024). A combination of both techniques can be realized by first sampling an initial training set on diverse topics and then selecting from this sample stance-similar instances for an input instance. Such a careful selection of few-shots requires datasets that are diverse and representative in terms of topic and stance. A first investigation of the topic distribution of existing argument corpora can be found in the work of Ajjour et al. (2023).

7 Conclusion

In this paper, we have investigated what makes a set of training instances effective in few-shot stance classification. By modeling the task in an instance-specific way, we have proposed two alternative priming strategies: one that retrieves semantically similar training instances to the target instance and one that retrieves instances with a similar stance to it. We have utilized the training instances as few-shots both in a prompting approach and by instruction fine-tuning the LLMs.

Our experiments on three datasets demonstrate the effectiveness of the priming strategies when compared to choosing random or diverse instances for two models, Llama2-7b and Vicuna-7b. They also suggest that the priming effect is larger in prompting than in instruction fine-tuning. In addition to advancing the state of the art on stance classification, our work gives indications on the extent to which LLMs can be affected by priming. It also provides evidence that consistency among the training instances and between the training and the test instance is an important property of effective few-shots in prompting LLMs.

Future research may investigate more informed ways to sample effective priming instances (e.g., using meta-learning). While retrieval strategies have yielded promising effectiveness in our experiments, their success is bound to the availability of comprehensive training datasets. In case of data scarcity (indicated by the low similarity of the retrieved instances), generating priming instances for an input instance is a fruitful research direction to follow.

8 Limitations

In this paper, we have explored priming strategies for few-shot stance classification that take the semantic similarity and stance similarity between arguments into account. One of the limitations of the study is that we fixed the order of the instances for all priming strategies. In our experiments, we sorted the sampled instances alphabetically by their topics in all settings. This factored out the effect of the order of the instances on the effectiveness of a model. The gained comparability comes at the cost of guiding the order of the instances in a more supervised way.

Another limitation of our priming strategies is the incurred cost of computation for the instruction fine-tuning approaches. For example, the strategy `semantic-priming` samples for each argument those instances that are most semantically similar and then fine-tunes Alpaca-7b or Mistral-7b-instruct on this subset. This increases the computational complexity of the approach, since fine-tuning for each test argument takes notable time. Running Alpaca-7b or Mistral-7b-instruct on the VAST dataset took 16 GPU hours on NVIDIA A100. Nevertheless, we expect future approaches to these problems to be more efficient by speeding up the optimization process or applying techniques such as continual learning.

References

Yamen Ajjour, Johannes Kiesel, Benno Stein, and Martin Potthast. 2023. Topic Ontologies for Arguments. In *17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*. Association for Computational Linguistics.

Emily Allaway and Kathleen McKeown. 2020. Zero-shot stance detection: A dataset and model using generalized topic representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Erik Arakelyan, Arnav Arora, and Isabelle Augenstein. 2023. Topic-guided sampling for data-efficient multi-domain stance detection.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. *CoRR*, abs/1606.05464.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261.

Joe Barrow, Rajiv Jain, Nedim Lipka, Franck Dernoncourt, Vlad Morariu, Varun Manjunatha, Douglas Oard, Philip Resnik, and Henning Wachsmuth. 2021. Syntopical graphs for computational argumentation tasks. 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 1583–1595. Association for Computational Linguistics.

Tilman Beck, Andreas Waldis, and Iryna Gurevych. 2023. Robust integration of contextual information for cross-target stance detection. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (* SEM 2023)*, pages 494–511.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing things from a different angle: Discovering diverse perspectives about claims.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Stefano DellaVigna and Ethan Kaplan. 2007. The fox news effect: Media bias and voting. *The Quarterly Journal of Economics*, 122(3):1187–1234.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. 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)*, pages 3816–3830, Online. Association for Computational Linguistics.

Hans W. A. Hanley and Zakir Durumeric. 2023. TATA: stance detection via topic-agnostic and topic-aware embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 11280–11294. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Shanto Iyengar and Kyu S Hahn. 2009. Red media, blue media: Evidence of ideological selectivity in media use. *Journal of communication*, 59(1):19–39.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yan Jiang, Jinhua Gao, Huawei Shen, and Xueqi Cheng. 2022. Few-shot stance detection via target-aware prompt distillation. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 837–847.

Jaap Jumelet, Willem Zuidema, and Arabella Sinclair. 2024. Do language models exhibit human-like structural priming effects? *arXiv preprint arXiv:2406.04847*.

Rinaldo Kuehne, Christian Schemer, Joerg Matthes, and Werner Wirth. 2011. Affective priming in political campaigns: How campaign-induced emotions prime political opinions. *International Journal of Public Opinion Research*, 23:485–507.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. volume abs/2212.06800.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*.

Rui Liu, Zheng Lin, Yutong Tan, and Weiping Wang. 2021. Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3152–3157.

David Meyer and Roger Schvaneveldt. 1971. Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of experimental psychology*, 90:227–34.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kanishka Misra, Allyson Ettinger, and Julia Rayz. 2020. Exploring BERTs sensitivity to lexical cues using tests from semantic priming. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4625–4635, Online. Association for Computational Linguistics.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Myrthe Reuver, Suzan Verberne, and Antske Fokkens. 2024. Investigating the robustness of modelling decisions for few-shot cross-topic stance detection: A preregistered study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 9245–9260.

Benjamin Schiller, Johannes Daxenberger, Andreas Waldis, and Iryna Gurevych. 2024. Diversity over size: On the effect of sample and topic sizes for topic-dependent argument mining datasets. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 10870–10887. Association for Computational Linguistics.

Tamir Sheafer. 2007. How to evaluate it: The role of story-evaluative tone in agenda setting and priming. *Journal of communication*, 57(1):21–39.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 226–234, Suntec, Singapore. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8968–8975.

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`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Penghui Wei and Wenji Mao. 2019. Modeling transferable topics for cross-target stance detection. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1173–1176. ACM.

Bowen Zhang, Jun Ma, Xianghua Fu, and Genan Dai. 2025. Logic augmented multi-decision fusion framework for stance detection on social media. volume 122, page 103214.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

9 Appendix

Hardware We ran our experiments on NVIDIA A100 with 80 GB. The instance has eight 8-core CPU, each of which has access to 32 GB RAM.

Implementation Details We used the following models on Hugging Face in our experiments:

- Llama-2-7b-hf
- Alpaca-7b
- Mistral-7b-instruct-v0.2
- Vicuna-7b-v1.5

For optimizing the models, we used grid-search on the hyperparameters in Table 6. The hyperparameter values we used to optimize DeBERTa are listed in Table 5. As an optimizer for our models we always used AdamW.

Ethical Considerations Our research illustrates that systematically using instances with a certain stance in the prompt entices certain models to output content with a consistent stance. We do not see any ethical consequences of our research, given that we simply explored the behavior of existing LLMs under such priming. However, we stress that priming can be used maliciously by injecting polarized content in the prompt to force the model to generate a certain output. In particular, we point to two aspects here:

First, while safeguards might suppress direct malicious prompts, priming can be used to steer the model to invoke the generation of certain outputs in an implicit way. The consequences of priming might be more decisive for high-stakes tasks such as content moderation, where certain content is filtered. Hence, detecting and countering malicious usages of priming is an important research direction in the area of LLM safety.

And second, malicious usages can also inject associated priming instances and targets in the training data to increase the chances of certain associations later by the LLM. Linking priming instances and targets as preparation for priming allows even higher control over the output of the model. Detecting and filtering such injected associations is an open research challenge, given the sheer size of data that is used for pre-training LLMs.

Given are the following arguments
On the topic `{training topic}`, the argument `{training argument}` has the stance `{training stance}`.
Classify the stance of the following argument on the given topic into Pro or Con:
On the topic `{test topic}`, the argument`{test argument}` has the stance

Table 2: The template for the few-shot stance classification using prompt-based methods. The second line stands for the few-shot instances and is populated with the sampled instances only in prompting. Notice that for Alpaca-7b, we change how instances are formatted to adhere to its template.

Dataset	**Split**	**Instances**	**Topics**	**Pro**	**Con**	**Neutral**
VAST	Training	13,477	4,641	5,327	5,595	2,555
	Validation	1,019	389	321	350	348
	Test	1,460	600	451	490	519
IBMSC	Training	604	10	340	264	-
	Validation	435	15	285	150	-
	Test	1,355	30	700	655	-
Perspectrum	Training	6,978	541	3,599	3,379	-
	Validation	2,071	139	1,047	1,024	-
	Test	2,773	227	1,471	1,302	-

Table 3: Distribution of instances across VAST, IBMSC, and Perspectrum datasets.

Hyperparameter	**IBMSC**	**VAST**	**Perspectrum**
Batch size	8	64	8
Epochs	1	15	15
Learning rate	10^{-4}	10^{-5}	10^{-5}

Table 4: Hyperparameters for DeBERTa for the datasets: IBMSC, VAST, and Perspectrum.

Hyperparameter	**Value**
Batch size	$[4, 8, 16, 32, 64]$
Learning rate	$[10^{-4}, 10^{-5}, 3 \times 10^{-5}, 10^{-6}, 10^{-7}]$

Table 5: The value range for each hyperparameter used to optimize DeBERTa. candidates.

Hyperparameter	**Value**
Batch size	$[4, 8, 16, 32, 64]$
Learning rate	$[10^{-3}, 10^{-4}, 3 \times 10^{-4}, 10^{-5}, 2 \times 10^{-5}, 10^{-6}, 10^{-7}]$
Early stopping threshold	$[10^{-1}, 2 \times 10^{-1}, 3 \times 10^{-2}, 3 \times 10^{-4}, 10^{-5}, 2 \times 10^{-5}, 10^{-6}, 3 \times 10^{-7}]$

Table 6: The value range for each hyperparameter used to optimize Alpaca-7b and Mistral-7b-instruct.

	IBMSC		**VAST**		**Perspectrum**	
Hyperparameter	Alpaca	Mistral	Alpaca	Mistral	Alpaca	Mistral
Batch size	4	8	64	32	4	32
Epochs	140	50	50	110	110	110
Learning rate	3×10^{-4}	2×10^{-4}	5×10^{-5}	2×10^{-4}	3×10^{-4}	2×10^{-4}
Early stopping	1	1	1	1	1	1
Early stopping threshold	0	5×10^{-2}	10^{-2}	3×10^{-7}	3×10^{-7}	10^{-6}
Warmup steps	100	100	100	100	100	100
Cutoff len	256	8192	2048	8192	2048	8192
Lora rank	8	8	8	8	8	8
Lora dropout	0.05	0.05	0.05	0.05	0.05	0.05
Lora alpha	16	16	16	16	16	16

Table 7: Hyperparameters for Alpaca-7b and Mistral-7b-instruct models across the three datasets.