Target and Stance Generation using Finetuned Llama for Open-Target Stance Detection

April 25, 2025

Abstract

Open-Target Stance Detection (OTSD) requires identifying a target within a text and determining the expressed stance without prior knowledge of potential targets during inference, unlike traditional stance detection where targets are predefined. This study explores finetuning the Llama 3.1 8B model for OTSD using Parameter-Efficient Finetuning (PEFT) with Low-Rank Adaptation (LoRA) on a combined dataset from TSE and VAST resources. We evaluated the finetuned model against its base counterpart on three datasets: COVID-19 (specialized domain), EZStance Mixed (general-domain), and EZStance Noun Phrase (noun phrase targets). Semantic assessment by Gemini and DeepSeek-671B, averaged for robustness, and BERTweet-based similarity confirmed the finetuned model's superior performance, particularly on COVID-19, achieving 66.30% stance accuracy and 52.47% target accuracy (vs. 45.28% and 25.39% for the base model). EZStance Mixed showed moderate gains, while EZStance Noun Phrase exhibited significant target accuracy improvement but a slight stance accuracy decline, underscoring domain-specific challenges. These results highlight the efficacy of finetuning for OTSD.

Keywords: Open-Target Stance Detection, Llama, Finetuning, LoRA, Stance Detection, Target Generation, Parameter-Efficient Finetuning, Semantic Evaluation.

1 Introduction

Stance Detection (SD) is a fundamental task in natural language processing, focusing on determining the viewpoint (e.g., favor, against, neutral) expressed in a text towards a given entity or topic, known as the target [1]. Many real-world applications, such as social media analysis, opinion polling, and argument mining, benefit from accurate stance detection. However, a significant limitation of many existing SD approaches is the requirement that the target of the stance be provided beforehand. This assumption does not hold in many practical scenarios where the target itself needs to be identified from the text.

This leads to the more complex and realistic task of Open-Target Stance Detection (OTSD), where the model must first identify the relevant target(s) within the text and then determine the stance towards that identified target [1]. The OTSD task presents challenges in both target generation (TG) and subsequent stance classification, especially when targets are not explicitly mentioned. While Zero-Shot Stance Detection (ZSSD) methods have emerged to handle targets unseen during training [4], they typically still require the target to be provided during inference. Even recent approaches like Target-Stance Extraction (TSE) [**?**], which generate targets, often rely on mapping these generated targets to a predefined list, limiting their truly "open" nature.

Recent advancements in Large Language Models (LLMs) have shown promise in various zero-shot and few-shot learning scenarios. Their ability to understand context and generate text makes them potential candidates for tackling the complexities of OTSD. Initial work by Akash et al. [1] explored the zero-shot capabilities of various LLMs, including Llama, for OTSD, highlighting their potential but also the challenges, particularly with non-explicit targets.

This paper investigates the effectiveness of *finetuning* an LLM for the OTSD task. Specifically, we focus on the Llama 3.1 8B architecture [2] and employ

Parameter-Efficient Finetuning (PEFT) using Low-Rank Adaptation (LoRA) [3]. We train the model on a combined dataset derived from existing stance detection resources (TSE and VAST datasets, as used in [1]) to simultaneously generate the target and predict the stance in a specific format. We evaluate the performance of the finetuned model against its base zero-shot counterpart on three distinct test datasets (COVID-19, EZStance Mixed, EZStance Noun Phrase) using semantic evaluation performed by state-of-the-art LLMs (Gemini and DeepSeek-671B). Our contribution lies in demonstrating the impact of task-specific finetuning via LoRA on the Llama model's ability to address the joint target generation and stance detection challenges inherent in OTSD, with particularly strong improvements in specialized domains like COVID-19 and more moderate gains in general-domain datasets.

2 Literature Review

Research in Stance Detection (SD) has evolved significantly. Early work focused on supervised classification tasks where both the text and the target were provided, often using feature engineering and traditional machine learning models, later transitioning to deep learning approaches like LSTMs and CNNs [4].

Recognizing the impracticality of gathering labeled data for every conceivable target, Zero-Shot Stance Detection (ZSSD) emerged. ZSSD aims to predict stances towards targets unseen during training. Various techniques have been explored, including contrastive learning, generative approaches, prompt-based learning with LLMs, and methods incorporating external knowledge [4]. A prominent line of work, TSE [**?**, **?**], generates targets but subsequently maps them to a known target list, facilitating evaluation but deviating from a truly open-target scenario.

The specific challenge of Open-Target Stance Detection (OTSD), where the target is neither provided nor drawn from a predefined inference-time list, was formally introduced and benchmarked by Akash et al. [1]. Their work evaluated the zero-shot performance of several state-of-the-art LLMs (GPT-3.5, GPT-4o, Mistral) on OTSD, establishing baseline capabilities and highlighting the difficulty posed by implicitly mentioned targets. They demonstrated that while LLMs outperform prior methods in target generation quality, significant challenges remain.

Contemporaneously, the capabilities of Large Language Models (LLMs) like Llama [2], GPT, and Mistral have rapidly advanced. Their success in diverse generative and instruction-following tasks suggests their potential for the joint generation requirement of OTSD.

However, fully finetuning such large models is computationally expensive. Parameter-Efficient Finetuning (PEFT) methods have been developed to mitigate this. Among these, Low-Rank Adaptation (LoRA) [3] is a popular technique that introduces trainable low-rank matrices into the existing layers of an LLM, allowing adaptation with significantly fewer trainable parameters compared to full finetuning.

This work builds directly upon the OTSD task defined by Akash et al. [1]. Instead of evaluating zero-shot performance, we investigate the impact of applying PEFT (specifically LoRA) to finetune a Llama 3.1 8B model explicitly for the combined target generation and stance classification task inherent in OTSD.

3 Methodology

Our approach to Open-Target Stance Detection (OTSD) involves adapting a pretrained Large Language Model to jointly generate the stance target and classify the stance expressed towards it within a given input text.

3.1 Problem Definition

Let the training dataset be represented as $D = \{(x_i, t_i, y_i)\}_{i=1}^{N}$, where x_i denotes the input text, t_i is the corresponding ground truth target, and y_i is the ground truth stance label (e.g., "FAVOR", "AGAINST", "NONE"). The test dataset is represented similarly as $D' = \{(x'_j, t'_j, y'_j)\}_{j=1}^{M}$, where x'_j represents an input text potentially containing novel targets not seen in D.

The core objective in Open-Target Stance Detection (OTSD) is to train a model M that, given an input text x', generates an output sequence O encompassing

both a predicted target $\hat{t}'$ and a predicted stance $\hat{y}'$. Formally:

$$O = M(x')$$

The generated output sequence O must be parsable to extract the predicted target $\hat{t}'$ and stance $\hat{y}'$. In our implementation, the model is trained using supervised finetuning on D to produce outputs formatted precisely according to a predefined template:

```
Below is an instruction that describes a
    task, paired with an input that
    provides further context. Write a
    response that appropriately completes
    the request.

### Instruction:
Analyze the stance and identify the target
    in the following text.

### Input:
[Input Text]

### Response:
Target: [Target], Stance:
    [Stance]<|end_of_text|>
```

During inference, the `[Target]` and `[Stance]` parts are left empty for the model to generate. The goal is to train M such that for unseen inputs x' from the test distribution D', the generated $\hat{t}'$ closely approximates the ground truth target t' and the predicted stance $\hat{y}'$ matches the ground truth stance y'.

3.2 System Overview

Figure 1 presents the high-level architecture of our OTSD system. The pipeline consists of three main components:

- **Input Processing**: The system accepts raw text input from various sources such as social media posts and news comments.
- **Model Processing**: A finetuned Llama 3.1 8B model with LoRA adaptations processes the input, optimized using 4-bit quantization and the Unsloth library.
- **Structured Output**: The model generates structured data containing the identified target and corresponding stance.

3.3 Model Architecture

The foundation of our approach is the **Llama 3.1 8B** model, a highly capable Large Language Model (LLM) developed by Meta [2]. As a member of the Llama family, Llama 3.1 8B represents the cutting edge in open foundation models, pretrained on trillions of tokens from a diverse range of publicly available sources. Architecturally, it follows the standard **transformer-based, decoder-only** paradigm. This means the model processes input sequences and autoregressively generates output tokens one after another, conditioning each new token on the preceding sequence. This inherent generative capability makes it well-suited for tasks requiring text production, such as the joint target and stance generation needed for OTSD. The 8B parameter variant strikes a balance between strong performance and manageable computational requirements for finetuning and inference. For our experiments, we specifically utilized the `unsloth/Meta-Llama-3.1-8B` version available on Hugging Face, incorporating optimizations provided by the Unsloth library [5].

While the base Llama 3.1 8B model possesses extensive world knowledge and language understanding abilities from its pretraining, achieving optimal performance on a specific, nuanced task like OTSD necessitates task-specific adaptation. Fully finetuning all 8 billion parameters, however, demands substantial computational resources (GPU memory and time) often unavailable in typical research settings. To overcome this limitation, we employed **Parameter-Efficient Finetuning (PEFT)**, a family of techniques designed to adapt large pretrained models by modifying only a small fraction of their parameters.

Specifically, we adopted **Low-Rank Adaptation (LoRA)** [3]. LoRA operates on the principle that the change in weights required to adapt a pretrained model to a new task often has a low intrinsic rank. Instead of learning the full change ΔW for a weight matrix W, LoRA learns its low-rank decomposition $\Delta W = BA$, where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ are signif-

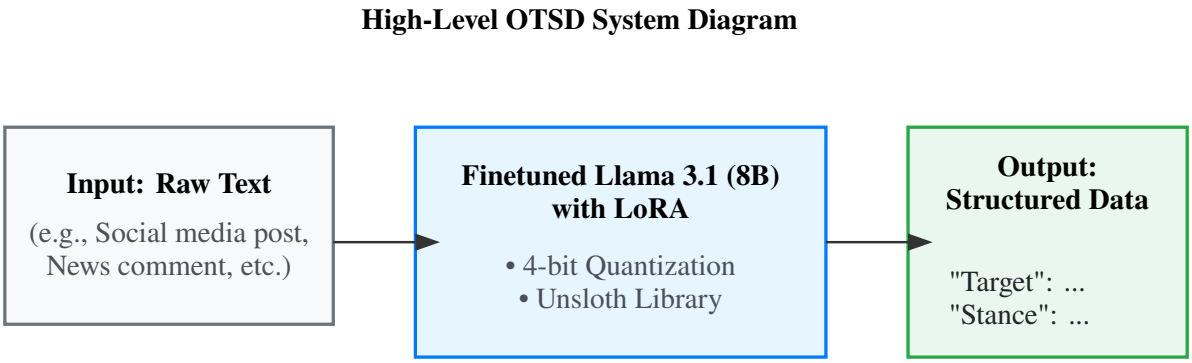

Figure 1: High-Level OTSD System Diagram showing the complete pipeline from raw text input to structured stance and target output. The system utilizes a finetuned Llama 3.1 8B model with LoRA adaptations, optimized using 4-bit quantization and the Unsloth library.

icantly smaller matrices (the "LoRA adapters") with rank $r \ll \min(d, k)$. During finetuning, the original pretrained weights W are kept frozen, and only the parameters within the newly introduced matrices A and B are updated via gradient descent. The effective weight matrix during forward passes becomes $W + BA$. This approach dramatically reduces the number of trainable parameters; in our setup, only approximately 0.52% of the total model parameters were trained.

The specific configuration of LoRA in our experiments, guided by the implementation in the accompanying notebook and common practices for effective adaptation, included:

- **Rank (r):** Set to **16**. This determines the inner dimension of the LoRA matrices A and B, controlling the capacity of the adaptation.
- **LoRA Alpha (α):** Set to **16**. This acts as a scaling factor for the LoRA update. The effective update applied to the forward pass is $(\alpha/r) \times BA$. Setting $\alpha = r$ is a common initialization strategy.
- **Targeted Modules:** LoRA adapters were strategically injected into specific layers crucial for language processing within the Llama architecture. These included the query (`q_proj`), key (`k_proj`), value (`v_proj`), and output (`o_proj`) projection layers within the multi-head self-attention mechanism, as well as the gate (`gate_proj`), up (`up_proj`), and down (`down_proj`) projection layers within the feed-forward network blocks. These modules were selected as they are critical for adapting the attention and feed-forward components of the transformer architecture.
- **LoRA Dropout:** Set to 0, disabling dropout within the LoRA layers for optimized performance as suggested by Unsloth.
- **Bias:** Set to `"none"`, indicating that bias terms within the LoRA adapters were not trained.

To facilitate this finetuning process on resource-constrained hardware (specifically, an NVIDIA T4 GPU with approx. 15GB VRAM as used in the notebook environment), we leveraged the **Unsloth library** [5]. Unsloth provides optimized implementations for faster training and reduced memory usage. Key techniques enabled by Unsloth in our setup were:

- **4-bit Quantization:** The base Llama 3.1 8B model weights were loaded in 4-bit precision (`load_in_4bit = True`) using bitsandbytes [**?**]. This technique significantly compresses the model size, reducing the GPU memory required to load the model.
- **Gradient Checkpointing:** We utilized Unsloth's optimized implementation of gradient checkpointing (`use_gradient_checkpointing = "unsloth"`). This technique avoids storing all intermediate activations during the forward pass, instead recomputing them during the backward pass, drastically reducing memory usage at the

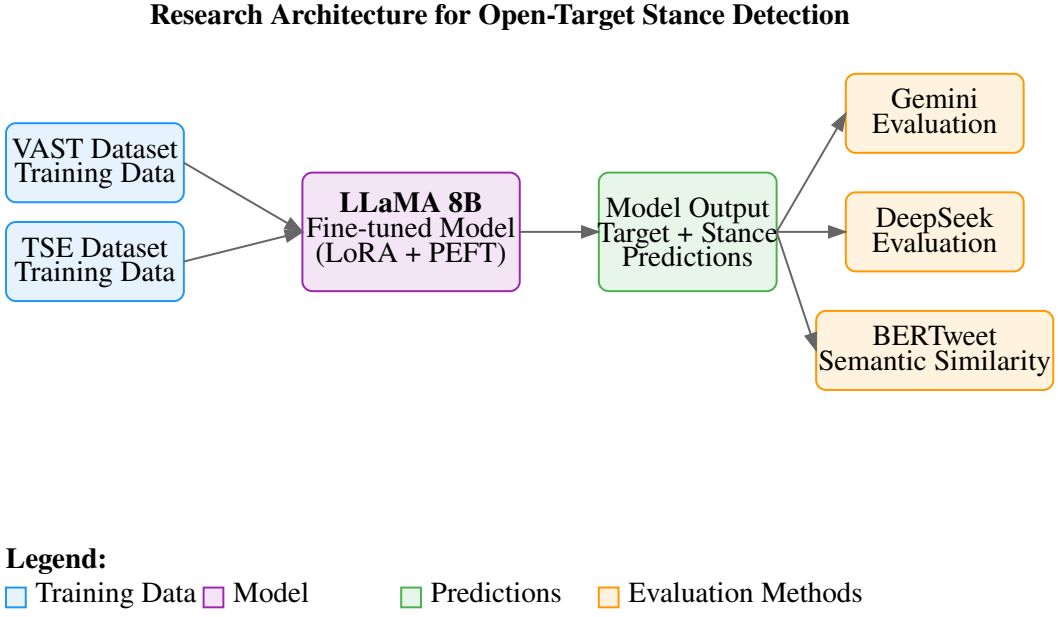

Figure 2: Detailed Research Architecture showing the complete pipeline from training data through model finetuning to evaluation. The system combines VAST and TSE datasets for training, processes through a finetuned Llama 3.1 8B model, and evaluates outputs using multiple methods including Gemini, DeepSeek, and BERTweet for semantic similarity assessment.

cost of increased computation time, enabling training with a sequence length of 2048 tokens.

The resulting model, incorporating the frozen Llama 3.1 8B base weights and the trained LoRA adapters, constitutes our finetuned OTSD model ($M_{\text{finetuned}}$). This architecture is compared against the performance of the base 4-bit quantized Llama 3.1 8B model (M_{base}) used directly for the task without LoRA adapters or finetuning.

3.4 Datasets

Our work utilizes several datasets for finetuning and evaluation, covering diverse domains and stance targets:

3.4.1 Finetuning Dataset

For finetuning the Llama 3.1 8B model, we constructed a combined dataset by merging resources from established stance detection benchmarks:

- **TSE Dataset** [7]: Focuses on Target-Stance Extraction, providing examples with explicitly mentioned targets and stances. It includes a variety of social media texts with labeled targets and stances.
- **VAST Dataset** [8]: Offers a variety of topics and stances, contributing to the diversity of the training data. It covers multiple domains, enhancing the model's ability to generalize.

This combined dataset, comprising 6,480 examples (as per the notebook logs), was used exclusively for the PEFT process. Each entry contained the input text,

the ground truth target, and the ground truth stance (FAVOR, AGAINST, or NONE), formatted into the instruction-following template described in Section 3.1.

Importantly, the training and test datasets contain no common texts, underscoring the open stance target detection nature of this study.

Table 1 presents representative examples from our training dataset, showcasing the diversity of topics and stance expressions.

3.4.2 Evaluation Datasets

To rigorously evaluate the performance and generalization capabilities of both the base and finetuned models, we employed three distinct test datasets, separate from the finetuning data. Table 2 presents representative examples from each test dataset.

- **COVID-19**: A dataset focused specifically on stances related to the COVID-19 pandemic, representing a specialized domain with domain-specific terminology and sentiment.
- **EZStance Mixed** [1]: A general-domain dataset containing a mix of topics and targets, testing the model's ability to handle diverse inputs.
- **EZStance Noun Phrase** [1]: Focuses on noun phrases as targets, presenting a unique linguistic challenge that may affect stance classification due to the abstract or implicit nature of targets.

These datasets served as the held-out test sets for evaluating both the base and finetuned models using the semantic evaluation methodology outlined in Section 4.

3.5 Finetuning Details

The finetuning process was carried out using the Supervised Finetuning Trainer (`SFTTrainer`) from the Hugging Face TRL library, integrated with the Unsloth optimizations. The training objective was standard causal language modeling loss, where the model learns to predict the next token in the sequence, specifically trained to generate the `Target: ..., Stance: ...` response based on the provided instruction and input text.

Key hyperparameters for the training process were set as follows (based on the notebook implementation):

- Maximum Sequence Length: 2048 tokens.
- Per Device Train Batch Size: 2.
- Gradient Accumulation Steps: 4 (resulting in an effective batch size of 8).
- Learning Rate: 2×10^{-4}.
- Optimizer: AdamW 8-bit [**?**].
- Learning Rate Scheduler: Linear decay.
- Warmup Steps: 5.
- Maximum Training Steps: 60.
- Floating Point Precision: BF16 enabled if supported, otherwise FP16.
- Seed: 3407 for reproducibility.

Training was performed on a single NVIDIA T4 GPU.

3.6 BERTweet Semantic Evaluation

In addition to DeepSeek and Gemini evaluations, we employed BERTweet [6] for semantic similarity assessment of predicted targets against ground truth. BERTweet provides a specialized evaluation metric for measuring semantic similarity in the context of social media text, which is particularly relevant for our stance detection task. Separately, we calculated Macro F1 scores to evaluate the stance classification performance across the FAVOR, AGAINST, and NONE categories.

The BERTweet evaluation results and Macro F1 scores (Table 4) reveal interesting patterns:

- For the COVID-19 dataset, the finetuned model achieved higher semantic similarity (0.7832) compared to the base model (0.7363), and a higher Macro F1 score for stance classification (0.2947 vs. 0.2660).

Table 1: Sample Training Data Points

Target	**Text**	**Stance**
technology	Necessity is the mother of all invention. Never bet against human ingenuity, especially when survival is on the line. The technology we need to move beyond fossil fuels exist today - wind, solar, efficiency improvements like LED lighting, and especially nuclear power.	AGAINST
metoo movement	Other ways to heal: when you can't tell your #MeToo story. The technology we need to move beyond fossil fuels exist today.	AGAINST
merger of disney and fox	COMCAST willing to outbid Disney for Fox... could the deal between Disney and Fox be over?	AGAINST

Table 2: Sample Test Data Points from Different Evaluation Datasets

Target	**Text**	**Stance**
COVID-19 Dataset		
face masks	Maybe if people want to walk around without masks on they can organize into fenced-off private mask nudist colonies, keep their idiotic sense of liberty, and the rest of us can live in a society. #WearAMask	FAVOR
EZStance Mixed Dataset		
Trump-appointed judge	WhiteHouse has appealed a Trump-appointed judge's obstruction of CDCgov's transportation mask mandate, but without an emergency stay, millions of lives remain in danger as we work our way through the courts.	AGAINST
EZStance Noun Phrase Dataset		
Cummings	Fgs stop calling Cummings a mad genius. He's just mad. The higher echelons of the civil service are as hidebound by private education as the Tory party, law and the BBC.	AGAINST

- On the EZStance Mixed dataset, the finetuned model demonstrated improved performance with higher semantic similarity (0.7779) compared to the base model (0.7392), and a better Macro F1 score (0.3964 vs. 0.3624).

- Similarly, for the EZStance Noun Phrase dataset, the finetuned model showed superior results with higher semantic similarity (0.7871) compared to the base model (0.7390), and a better Macro F1 score (0.3907 vs. 0.3533).

These results consistently demonstrate that finetuning improves performance across all datasets, with particularly strong improvements in both semantic similarity and stance classification for specialized domains like COVID-19, as well as notable gains in general-domain datasets. This reinforces our findings about the effectiveness of the finetuning approach for both target identification and stance classification tasks.

4 Results and Discussion

We evaluated our models on three distinct datasets to assess their generalization capabilities and performance across different domains. The results are presented for both the base Llama model (M_{base}) and the finetuned model ($M_{\text{finetuned}}$), with semantic evaluation performed by both DeepSeek-671B and Gemini. Table 3 presents the comprehensive evaluation results across all datasets.

4.1 Comparison with Previous Work

The comparison reveals several key insights:

- For EZStance Mixed, our model achieves better overall performance compared to previous zero-shot approaches

- In the EZStance Noun Phrase dataset, while target identification improved significantly, stance detection showed mixed results compared to previous work.

4.2 Analysis of Results

Several key observations emerge from these results:

- **COVID-19 Dataset Performance**: The most substantial improvements were observed in the COVID-19 dataset, where the finetuned model achieved:
  - An average stance accuracy increase of 21.02 percentage points (from 45.28% to 66.30%)
  - An average target accuracy increase of 27.08 percentage points (from 25.39% to 52.47%)

- **EZStance Mixed Dataset**: Significant improvements were seen across all metrics:
  - Average stance accuracy improved by 5.24 percentage points
  - Average target accuracy improved by 7.30 percentage points
  - BERTweet semantic similarity increased from 0.7392 to 0.7779
  - Macro F1 score improved from 0.3624 to 0.3964

- **EZStance Noun Phrase Dataset**: Consistent improvements across all metrics:
  - Average target accuracy increased by 8.28 percentage points
  - BERTweet semantic similarity improved from 0.7390 to 0.7871
  - Macro F1 score increased from 0.3533 to 0.3907

- **Evaluator Consistency**: The averaged results from DeepSeek and Gemini demonstrate consistent trends, though with some variations:
  - DeepSeek typically showed higher stance accuracy scores
  - Gemini often provided higher target accuracy scores

Table 3: Comprehensive Evaluation Results Across All Datasets (%)

Dataset	Model	DeepSeek		Gemini		Average	
		Stance	Target	Stance	Target	Stance	Target
COVID-19	Base	47.22	25.57	43.33	25.21	45.28	25.39
	Finetuned	**71.43**	42.24	61.17	**62.69**	**66.30**	**52.47**
EZStance Mixed	Base	43.78	26.94	44.22	26.03	44.00	26.49
	Finetuned	**50.00**	**34.26**	48.48	33.31	**49.24**	**33.79**
EZStance Noun Phrase	Base	**48.99**	27.09	46.67	25.70	**47.83**	26.40
	Finetuned	47.13	**35.29**	47.11	34.07	47.12	**34.68**

Table 4: BERTweet Target Semantic Similarity and F1 Score Evaluation

Dataset	Model	BERTweet Similarity	Macro F1
COVID-19	Base	0.7363	0.2660
	Finetuned	**0.7832**	**0.2947**
EZStance Mixed	Base	0.7392	0.3624
	Finetuned	**0.7779**	**0.3964**
EZStance Noun Phrase	Base	0.7390	0.3533
	Finetuned	**0.7871**	**0.3907**

Table 5: Performance Comparison with Previous Work on EZStance Datasets

Dataset	Model	Target Generation	Stance Detection	Key Findings
EZStance Mixed	Finetuned Llama 3.1 8B	BERTweet: 0.7779 Target Acc: 33.79%	Macro F1: 0.3964 Stance Acc: 49.24%	Outperforms previous best (Mistral)
	Base Llama 3.1 8B	BERTweet: 0.7392 Target Acc: 26.49%	Macro F1: 0.3624 Stance Acc: 44.00%	Comparable to Akash et al.'s Llama-3
	GPT-4o (Akash et al.)	SS: 0.87 BTSD: 50.69	SC: 46.22%	Lower stance accuracy than our finetuned model
	Mistral (Akash et al.)	SS: 0.86 BTSD: 50.17	SC: 48.72%	Close but lower performance
EZStance NP	Finetuned Llama 3.1 8B	BERTweet: 0.7871 Target Acc: 34.68%	Macro F1: 0.3907 Stance Acc: 47.12%	Higher target accuracy but slightly lower stance accuracy
	Base Llama 3.1 8B	BERTweet: 0.7390 Target Acc: 26.40%	Macro F1: 0.3533 Stance Acc: 47.83%	Baseline for comparison
	GPT-4o (Akash et al.)	SS: 0.87 BTSD: 50.69	SC: 46.22%	Similar stance performance
	Mistral (Akash et al.)	SS: 0.86 BTSD: 50.17	SC: 48.72%	Slightly better stance detection

- The variation may stem from differences in their training data or architectural biases, with DeepSeek potentially better tuned for stance classification and Gemini for semantic target matching
- The averaged metrics provide a balanced view of model performance

These results demonstrate that our finetuning approach is particularly effective for specialized domains like COVID-19, while still providing meaningful improvements across more general datasets. The consistent improvement in target identification across all datasets validates the effectiveness of our approach for the core challenge of target extraction in OTSD. The averaged metrics from both evaluators provide a robust measure of performance, helping to mitigate potential biases from any single evaluation model.

5 Conclusion

This paper investigated the application of a finetuned Llama-based LLM to the Open-Target Stance Detection (OTSD) task, comparing a base Llama 3.1 8B model against a version finetuned using Low-Rank Adaptation (LoRA) on a combined TSE and VAST dataset. We evaluated performance using semantic assessment by state-of-the-art LLMs (DeepSeek-671B and Gemini) across three diverse test datasets.

Our findings demonstrate that finetuning via LoRA significantly enhances the model's ability to perform the joint task of target identification and stance classification compared to the base model in a zero-shot setting. The most substantial improvements were observed on the specialized COVID-19 dataset, with average target accuracy increasing by over 27 percentage points and average stance accuracy by over 21 percentage points. Moderate improvements were observed on the EZStance Mixed dataset, while the EZStance Noun Phrase dataset showed significant target accuracy gains but a slight decrease in stance accuracy, indicating domain-specific challenges.

These results confirm the potential of finetuned LLMs, particularly using efficient methods like LoRA, as a viable approach for addressing the complexities of OTSD. The use of semantic evaluation provides a more nuanced understanding of model capabilities than traditional metrics. Future work could explore longer finetuning durations, alternative PEFT strategies, and techniques to further enhance performance on general-domain stance classification. Overall, this study provides strong evidence for the utility of PEFT in adapting LLMs for the complex, generative demands of Open-Target Stance Detection, yielding notable improvements in both target and stance prediction accuracy as measured by semantic evaluation.

References

[1] Abu Ubaida Akash, Ahmed Fahmy, and Amine Trabelsi. Can Large Language Models Address Open-Target Stance Detection? *arXiv preprint arXiv:2409.00222*, 2024. https://arxiv.org/abs/2409.00222.

[2] Meta AI. Llama 3.1: Advancements in Open Foundation Models. *Meta AI Technical Report*, July 2024. https://ai.meta.com/blog/meta-llama-3-1/.

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR 2022)*. https://openreview.net/forum?id=nZeVKeeFYf9.

[4] Emily Allaway and Kathleen McKeown. 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 (EMNLP)*, pages 8913–8931, 2020. https://doi.org/10.18653/v1/2020.emnlp-main.717.

[5] Unsloth AI. Unsloth: Efficient Fine-Tuning for Large Language Models. *GitHub Repository*, 2024. https://github.com/unslothai/unsloth.

[6] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A Pre-trained Language Model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, 2020. https://doi.org/10.18653/v1/2020.emnlp-demos.2.

[7] TSE Dataset. Target-Stance Extraction Explicit Dataset. https://anonymous.4open.science/r/opentarget-5521/data/tse/tse_explicit.csv.

[8] VAST Dataset. VAST Filtered Examples Dataset. https://anonymous.4open.science/r/opentarget-5521/data/vast/vast_filtered_ex.csv.