Improving Dialect Robustness in Large Language Models via LoRA and Mixture-of-Experts

Sanjh Maheshwari*, **Aniket Singh Rajpoot***, **Oana Cocarascu, Mamta**
King's College London
{sanjhmaheshwari1209, aniket161200}@gmail.com
{oana.cocarascu, mamta.name}@kcl.ac.uk

Abstract

Despite the success of large language models (LLMs) in a wide range of applications, it has been shown that their performance varies across English dialects. Differences among English dialects are reflected in vocabulary, syntax, and writing style, and can adversely affect model performance. Several studies evaluate the dialect robustness of LLMs, yet research on enhancing their robustness to dialectal variation remains limited.

In this paper, we propose two parameter-efficient frameworks for improving dialectal robustness in LLMs: *DialectFusion* where we train separate LoRA layers for each dialect and apply different LoRA merging methods, and *DialectMoE* which is built on top of Mixture of Experts LoRA and introduces multiple LoRA-based experts to the feed-forward layer to internally model the dialectal dependencies. Our comprehensive analysis on five open-source LLMs for sentiment and sarcasm tasks in zero- and few-shot settings shows that our proposed approaches enhance the dialect robustness of LLMs and outperforms instruct and LoRA fine-tuning based approaches.

1 Introduction

Large Language Models (LLMs) have been successful in a variety of tasks including sentiment and sarcasm classification (Zhang et al., 2025; Vajjala and Shimangaud, 2025). However, these models often exhibit a bias towards mainstream dialects, limiting their peformance in dialect-specific contexts (Srirag et al., 2025b; Lin et al., 2025). Regional English exhibits distinct grammatical structures, vocabulary, and expressions. As LLMs are trained on datasets composed of standard English, they often struggle to generalize to more diverse linguistic varieties (Joshi et al., 2025).

Several studies have explored ways to improve dialect robustness using rule-based systems (Ziems et al., 2023) or adapting language models for dialectal tasks (Sun et al., 2023; Liu et al., 2023). More recently Srirag et al. (2025a) introduced BESSTIE, a manually annotated dataset with three varieties of English (Australian, Indian, British), however, their evaluation focused on encoder and decoder models. Although parameter-efficient methods such as LoRA (Hu et al., 2022) have been explored for dialect adaptation, existing studies (Faisal and Anastasopoulos, 2024; Liu et al., 2023) do not systematically evaluate their performance across modern LLMs nor investigate approaches to enhance robustness across dialect groups (Srirag et al., 2025a).

To mitigate these gaps, in this paper we propose two parameter-efficient approaches, *DialectFusion* and *DialectMoE*, to improve the dialectal robustness of open-source LLMs.[1] In *DialectFusion*, we train separate LoRA adapters for each dialect and then merge them using two methods: Learnable Concatenation (CAT) (Prabhakar et al., 2025) and TrIm, Elect, and Merge (TIES) (Yadav et al., 2023). In *DialectMoE*, LoRA-based experts are added to the feed-forward layer of an LLM along with a gate router (Li et al., 2024) to enhance dialect performance. We focus on three English dialects: Australian, British, and Indian.

To evaluate the effectiveness of our proposed approaches, we fine-tune five open-source LLMs: three predominantly pre-trained using English corpora (Mistral-v0.1 7B Instruct (Jiang et al., 2023), Gemma2-9B Instruct (Team, 2024), Phi-3 Medium Instruct (et al., 2024)) and two multilingual models (Llama 3.1 8B Instruct (Weerawardhena et al., 2025), Qwen 2.5 7B Instruct (Yang et al., 2025) using the BESSTIE dataset (Srirag et al., 2025a). We compare our approaches with instruction-tuned models and LoRA fine-tuning on the full dataset without dialect distinctions in zero- and few-shot

*Equal contribution.

[1]Our code is available at https://github.com/Sanjh-Maheshwari/LLM-Dialect-Robustness.

Proceedings of the Thirteenth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 293–303

settings. Our results show that our proposed methods consistently outperform both instruction-tuned and LoRA fine-tuning.

To summarize, our **contributions** are as follows:

- We introduce two parameter-efficient approaches based on LoRA adapters and Mixture-of-Experts to improve dialectal robustness in LLMs;
- We fine-tune and evaluate five recent LLMs on the BESSTIE dataset, covering sentiment and sarcasm classification, enabling a systematic assessment of robustness across English dialects;
- We compare our methods against strong baselines including instruction-tuned models and LoRA fine-tuning without dialect separation, and show consistent improvements in both zero-shot and few-shot settings, demonstrating improved robustness across dialects.

2 Related Works

Large Language Models (LLMs) often perform poorly on dialectal data due to syntactic, orthographic, and lexical variation (Srirag et al., 2025b). As training data is heavily skewed toward Standard American English (Joshi et al., 2025), monolingual and multilingual LLMs often fail to generalize to dialectal text without explicit adaptation.

Several efforts have been made to improve dialect robustness. Ziems et al. (2023) proposed Multi-Value, a rule-based translation system consisting of 50 English dialects and their 189 unique linguistic features. Using these transformation rules, Liu et al. (2023) created a synthetic dataset using transformation rules, trained adapters for each linguistic feature, and then fused these adapters into a single model. Some works adapted language models for tasks requiring extensive dialectal knowledge, however their focus was on smaller task-specific language models such as mT5, Flan-T5 rather than recent LLMs (Sun et al., 2023; Liu et al., 2023). In order to increase robustness to dialectal variance without impairing downstream task performance, Sun et al. (2023) proposed a dialect-robust evaluation metric and NANO, a training schema which introduces regional and language information to pretraining. Faisal and Anastasopoulos (2024) applied LoRA (Hu et al., 2022) to adapt a multilingual instruction-tuned model to improve task performance on three South Slavic dialects.

More recently, Srirag et al. (2025a) proposed a dataset for sentiment and sarcasm classification for three varieties of English: Australian, Indian, and British, collected from Google Places reviews and Reddit comments. They evaluated nine LLMs (6 encoders and 3 decoders) and showed that models are better at sentiment classification compared to sarcasm detection. They also highlighted that monolingual models perform slightly better than multilingual models suggesting that multilingual pre-training does not adequately capture intra-language variation. The results are reported only for fine-tuned models, which offers limited insights into the actual improvements on dialectal variants.

Previous work has not explored recent advances in LLM fine-tuning including LoRA merging strategies (e.g. CAT (Prabhakar et al., 2025) and TIES (Yadav et al., 2023)) and parameter efficient implementations (e.g. MixLoRA (Li et al., 2024)) despite their proven effectiveness in multi-task learning settings that closely resemble multi-dialect settings. We bridge this gap through a systematic evaluation of six methods across five LLMs in both zero-shot and few-shot settings.

3 Methodology

We propose two parameter-efficient approaches to improve dialect robustness in LLMs:

- *DialectFusion* trains separate LoRA adapters for each dialect and then merges them using two methods: CAT (Prabhakar et al., 2025) or TIES (Yadav et al., 2023).
- *DialectMoE* employs mixture-of-experts and inserts multiple LoRA-based experts within the feed-forward network blocks of the frozen pre-trained model with a top-k router to dynamically select dialect-specific experts without requiring separate adapter training (Li et al., 2024).

We focus on three English dialects, specifically Australian (AU), British (UK), and Indian (IN).

3.1 DialectFusion

DialectFusion is a parameter-efficient method designed to improve dialect robustness of LLMs. Here, we train multiple LoRA adapters, one for each target dialect, as shown in Figure 1. This allows the model to capture dialect specific linguistic features independently.

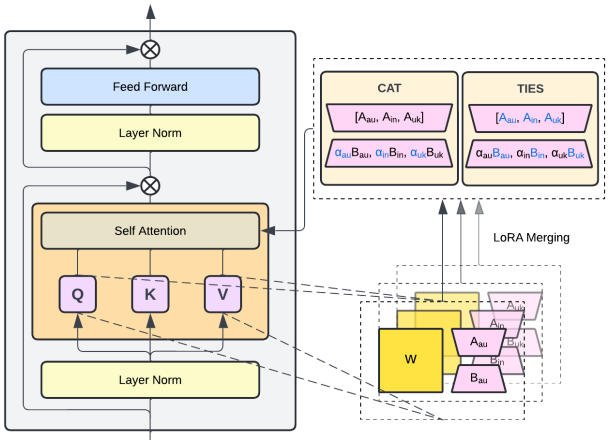

Figure 1: Overview of *DialectFusion* using LoRA merging (CAT and TIES). α_{au}, α_{in} and α_{uk} highlighted in blue in case of CAT are trainable. Similarly, A_{au}, A_{in}, A_{uk}, B_{au}, B_{in} and B_{uk} highlighted in blue in TIES are pre-trained dialect-specific LoRA adapters merged with fixed α weights.

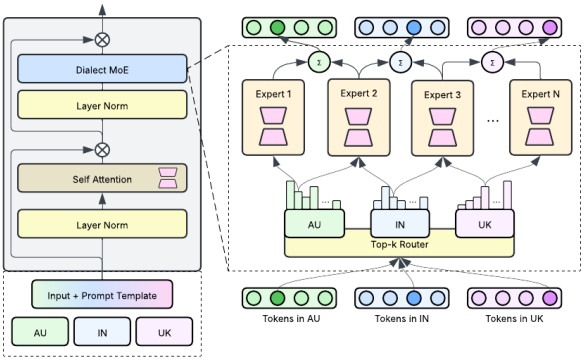

Figure 2: Overview of *DialectMoE*. LoRA adapters are applied to self-attention layers as standard. At the feed-forward layer, a sparse MoE is formed where all experts share the base Feed Forward Network weights.

One approach is to keep these adapters separate (LoRA Separate) where each dialect-specific adapter is loaded independently during inference based on the target dialect. While this maintains the distinct linguistic characteristics of each dialect, it requires prior knowledge of the dialect and separate inference passes for each dialect. Another approach is to merge the trained adapters using LoRA merging techniques such as CAT (Prabhakar et al., 2025) and TIES (Yadav et al., 2023) to produce a single adapter for all dialects.

CAT CAT is a LoRA merging method that first trains separate LoRA adapters for each dialect independently, then learns layer-specific (l) mixing coefficients to optimally combine these adapters, enabling the model to handle multiple dialects without retraining the base model. In each standard LoRA layer, the original weight matrix (W_0) stays frozen, while the update is calculated as $\Delta W = BA$, where $A \in \mathbb{R}^{r \times k}$ and $B \in \mathbb{R}^{d \times r}$ are low-rank decomposition matrices with $r \ll \min(d, k)$. Here, k and d are the input and output dimensions of W_0 respectively, and r is the low-rank bottleneck dimension. For an input vector x, the forward pass computes $W_0 x + BAx$.

CAT extends this by merging trained LoRA adapters. As shown in Figure 1, this approach combines LoRA updates from three dialect adapters using layer-specific mixing coefficients:

$$\Delta W^l = \alpha^l_{au} B_{au} A^\top_{au} + \alpha^l_{in} B_{in} A^\top_{in} + \alpha^l_{uk} B_{uk} A^\top_{uk}$$

where $\alpha^l_{au}, \alpha^l_{in}, \alpha^l_{uk} \in [0, 1]$ are trainable layer-specific merging coefficients for the Australian, Indian, and UK dialect adapters respectively, and A_k, B_k (for $k \in \{\text{au}, \text{in}, \text{uk}\}$) are the low-rank matrices from each pre-trained dialect-specific LoRA adapter. Unlike standard LoRA where only A and B are trained, in CAT the dialect-specific LoRAs are first trained independently, then frozen, and only the mixing coefficients α^l are learned to optimally combine them at each layer l.

TIES TIES addresses interference between merged models through a three-step process. This method applies pre-processing steps before merging. First, we prune the smallest values in (A_k, B_k) for $k \in \{au, in, uk\}$, retaining only the top $\lambda \in [0, 1]$ fraction based on magnitude. Second, we compute a majority sign mask by summing all pruned parameters and storing the sign. Finally, we apply weighted linear merging using only the parameters matching the majority sign, effectively resolving conflicts when merging adapters with overlapping parameters as shown below:

$$\begin{aligned}\Delta W^l = &(\alpha_{au} B_{au} + \alpha_{in} B_{in} + \alpha_{uk} B_{uk}) \\ &(\alpha_{au} A_{au} + \alpha_{in} A_{in} + \alpha_{uk} A_{uk})^\top\end{aligned}$$

The weights in the equation, α_{au}, α_{in}, and α_{uk} are fixed hyperparameters which are typically set uniformly, e.g., $\alpha_{au} = \alpha_{in} = \alpha_{uk} = 1/3$. (Yadav et al., 2023)

3.2 DialectMoE

DialectMoE inserts multiple LoRA-based experts within the feed-forward network blocks of the frozen pre-trained model, employing a top-k router mechanism to dynamically select the most relevant experts for each input (Li et al., 2024). In our case,

each expert is assumed to be specialized in a distinct dialect after training, allowing the model to process dialect-specific nuances without the need to train separate adapters for each dialect and overloading the GPU memory.

As shown in Figure 2, *DialectMoE* consists of two parts. The left component is the transformer architecture with frozen base model augmented with LoRA adapters at various layers (Self Attention, Layer Norm, and Feed-Forward layers). The right component shows the sparse MoE mechanism with three dialect-specific experts.

Input tokens first pass through the standard transformer layers (Self Attention and Layer Norm) in the base model. At the feed forward layers, the input tokens corresponding to each dialect are passed to the top-k router which computes routing probabilities and assigns each token from various tasks (i.e. in our case dialects: AU, IN, UK) to different expert modules. The boxes labeled "AU", "IN", and "UK" below the experts represent the weighted gating mechanism that determines how much each expert contributes to the final output.

Each expert module comprises a feed-forward network with LoRA adapters. The outputs from the selected adapters are then combined using weighted sum ($\sum$). The weights are determined using the routing probabilities assigned by the top-k router. The output is then passed to the next layers of transformers allowing the model to leverage dialect-specific knowledge while maintaining the efficiency of the sparse MoE architecture.

4 Experiments

Models We evaluate five open-source LLMs, three predominantly pre-trained using English corpora (Mistral-v0.1 7B Instruct (Jiang et al., 2023), Gemma2-9B Instruct (Team, 2024), Phi-3 Medium Instruct (et al., 2024)) and two multilingual models (Llama 3.1 8B Instruct (Weerawardhena et al., 2025), Qwen 2.5 7B Instruct (Yang et al., 2025). We conduct experiments in zero-shot and few-shot settings.

All models are fine-tuned in half-precision for 5 epochs, with a batch size of 8 and Adam optimizer. The optimal learning rate is highly dependent on the LLM as well as the technique, but varies from 1e-5 to 2e-4. Learning rate selection was based on monitoring training loss convergence during preliminary runs, with particular attention to the Mistral-v0.1 7B Instruct which required lower learning rates (1e-5) across all proposed approaches for stable training.

All experiments are performed using two NVIDIA A100 40GB GPUs and two NVIDIA A100 80GB GPUs. We implemented the proposed approaches using the MoE-PEFT library[2], an LLMOps framework designed for high-throughput fine-tuning, evaluation and inference.[3]

Dataset All experiments are conducted on BESSTIE, a dataset for sentiment and sarcasm classification crawled from Google Places reviews and Reddit comments (Srirag et al., 2025a). The dataset is labeled with three varieties of English: Australian (AU), British (UK), and Indian (IN).

Variety	Subset	Train	Valid	% Pos. Sent	% Pos. Sarcasm
AU	Google	946	130	73%	7%
	Reddit	1763	241	32%	42%
IN	Google	1648	225	75%	1%
	Reddit	1686	230	25%	13%
UK	Google	1817	248	75%	0%
	Reddit	1007	138	12%	22%

Table 1: Dataset statistics for the two tasks: sentiment and sarcasm classification.

Table 1 shows dataset statistics. BESSTIE comprises 11,963 samples across the three English dialects. The training set comprises 8,867 samples, the validation set 1,212 samples, and the test set 2,523 samples. The class distribution varies significantly, with the Google-sourced data indicating high positive sentiment (about 73–75%) and low sarcasm (0–7%). In contrast, the Reddit-sourced data for each type shows substantially lower positivity (varying from 12% to 32%) and greater sarcastic levels (13–42%). Overall, Google data tends to be more positive, less sarcastic, and more verbose, while Reddit data is more neutral or negative, more sarcastic, and more concise.

We utilize the provided training set for fine-tuning all models, while the validation set is used for evaluation purposes, as the test set is not included in the publicly available version.

Although both the Google and Reddit subsets are annotated for sentiment and sarcasm, the Google subset contains very few positive sarcasm instances across all dialects (0–7%) as evident from Table 1. Due to this extreme class imbalance, we do not conduct experiments on the sarcasm classification task using the Google data.

[2]https://github.com/TUDB-Labs/MoE-PEFT

[3]Further implementation details, including prompt templates, are in Appendix A.

Baselines As there are no baselines with a similar objective, we compare against the following:

- Zero-shot prompting: We utilise the instruction variant of LLMs using the same evaluation setup without any additional fine-tuning.
- Few-shot prompting: We give the instruction variant of LLMs two input-output examples in the prompt to demonstrate the task before evaluation, without additional fine-tuning.
- LoRA without dialect separation (LoRA Grouping): We train LoRA adapters by grouping the dialects for a subset of the data. For each model, we train a single adapter for the sarcasm classification task and two adapters for sentiment classification, one for each source (Google, Reddit).

5 Results and Analysis

The results for sarcasm and sentiment classification in zero-shot and few-shot settings are in Table 2 and Table 3, respectively. We report task performance using accuracy and weighted F_1 score. In general, sarcasm classification seems to be the more challenging task.

5.1 Sarcasm Classification

Which model performs best? In the zero-shot setting, Gemma-2 achieves the highest average performance across all dialectal variants, with an average F_1 score of 80.79% using LoRA Separate. In the few-shot setting, Llama 3.1 with *DialectMoE* achieves the highest average F_1 score of 72.47% across all dialectal variants, demonstrating substantial improvements, particularly on UK and IN dialects. Gemma-2 using LoRA Separate maintains strong performance with an average F_1 score of 76.20%.

How consistent is performance across dialects? Performance across dialects varies significantly. Australian English is the most challenging, with an F_1 score generally in the range of 40-75%. Indian English performs best when fine-tuned (70-80% F_1), but the baseline seems to struggle (10-40% F_1). UK English shows the highest variation in F_1 scores, ranging from 20 to 80% F_1).

How do proposed techniques compare against baselines? Baselines generally achieve low F_1 scores. For example, the results using Mistral-7B Instruct in zero-shot (33.34% on AU, 14.67% on UK, 9.99% on IN) are significantly lower than those of *DialectMoE* (66.29% on AU, 79.52% on UK, 81.0% on IN). This holds true for few-shot where the Mistral-7B Instruct performs worse (56.07% on AU, 25.88% on UK, 17.79% on IN) than LoRA Separate (63.82%, 71.71%, 80.27% on each dialect respectively).

Which merging technique is most effective in *DialectFusion*? LoRA Separate almost always outperforms CAT and TIES. It achieves the highest or the near-highest accuracy in almost all cases. In the zero-shot setting, CAT shows competitive performance especially for Phi-3 on UK and AU dialects, but TIES underperforms both LoRA Separate and CAT, often yielding results closer to the baseline.

How does *DialectFusion* compare to *DialectMoE*? *DialectMoE* excels for some models in zero-shot settings, achieving 5-15% F_1 improvements over *DialectFusion* methods, particularly on UK and IN dialects. DialectFusion (especially LoRA Separate) provides more consistent and reliable performance across both few-shot and zero-shot scenarios.

5.2 Sentiment Classification

Which model performs best? In the zero-shot setting, *DialectMoE* achieves remarkable performance on Reddit using Mistral 7B across dialects. For Google Sentiment, *DialectFusion* with CAT merging shows the strongest performance on Phi-3 and Qwen 2.5 models, with Qwen achieving 98.46% F_1 on AU and Phi-3 reaching 99.19% on UK dialect. In the few-shot setting, Gemma 2 with LoRA Grouping achieves the best performance on UK and AU on the Reddit subset, and maintains a high F_1 score on the Google subset.

How consistent is performance across dialects? Performance across dialects varies significantly. Indian English appears to be the most challenging dialect across sentiment tasks. On the Reddit subset, Mistral 7B shows the weakest results in zero-shot setting. For Google Sentiment, Llama 3.1 struggles across all dialects in the zero-shot setting. However, using *DialectMoE* shows substantial improvements, achieving scores above 85%. In both zero- and few-shot, UK English demonstrates the strongest and most stable overall performance, with scores typically ranging from 90-99% F_1 across different models and techniques.

Model	Technique	Reddit Sarcasm				Reddit Sentiment				Google Sentiment			
		AU	UK	IN	Avg	AU	UK	IN	Avg	AU	UK	IN	Avg
Mistral-7B	Instruct	33.34	14.67	9.99	19.33	77.51	80.23	73.58	77.11	93.85	94.74	83.60	90.73
	LoRA Group	43.63	68.38	**80.27**	64.09	88.38	94.20	83.44	88.67	96.92	96.32	86.52	93.25
	LoRA Sep	63.86	68.38	**80.27**	70.84	87.38	86.85	84.76	86.33	**97.68**	95.44	85.88	93.00
	CAT	43.63	68.38	**80.27**	64.09	85.61	88.97	83.71	86.10	97.68	96.72	84.76	93.05
	TIES	48.00	18.27	25.02	30.43	80.97	83.51	77.86	80.78	95.34	94.68	83.55	91.19
	DialectMoE	**66.29**	**79.52**	81.00	**75.60**	**90.91**	**96.66**	**90.44**	**92.67**	96.16	**97.16**	**88.16**	**93.83**
Phi-3	Instruct	72.69	44.82	58.53	58.68	86.85	89.54	83.11	86.50	94.76	95.64	86.47	92.29
	LoRA Group	57.36	70.22	77.89	68.49	87.90	**94.70**	85.68	89.43	94.59	96.70	86.99	92.76
	LoRA Sep	66.08	68.38	80.27	**71.58**	88.65	92.13	86.48	89.09	95.38	97.12	**87.76**	93.42
	CAT	50.33	**72.06**	**81.81**	68.07	**89.25**	91.52	**88.16**	**89.64**	**96.14**	**99.19**	87.55	**94.29**
	TIES	**73.41**	54.37	66.71	64.83	88.82	91.12	84.98	88.31	94.73	96.43	85.53	92.23
	DialectMoE	68.67	58.60	60.83	62.70	86.35	93.84	83.93	88.04	74.79	83.91	50.51	69.74
Qwen 2.5	Instruct	61.25	30.26	46.45	45.99	85.16	91.29	84.67	87.04	96.95	98.01	87.15	94.04
	LoRA Group	**74.67**	73.34	82.22	76.74	89.31	93.84	85.01	89.39	95.38	97.57	86.41	93.12
	LoRA Sep	**74.67**	73.34	82.22	76.74	89.31	93.84	85.01	89.39	95.38	97.57	86.41	93.12
	CAT	42.69	69.63	81.06	64.46	88.65	94.03	82.8	88.49	**98.46**	97.55	86.52	**94.18**
	TIES	59.71	70.44	76.5	68.88	85.16	93.1	86.48	88.25	94.69	97.61	85.69	92.66
	DialectMoE	73.17	**81.71**	**83.21**	**79.36**	**90.16**	**96.51**	**88.56**	**91.74**	94.59	**98.79**	**86.74**	93.37
Llama 3.1	Instruct	57.71	23.99	38.28	40.00	78.79	91.12	81.90	83.94	78.32	75.54	76.66	76.84
	LoRA Group	51.25	75.18	79.62	68.68	89.04	94.70	84.49	89.41	**97.68**	97.55	87.55	94.26
	LoRA Sep	61.02	68.38	80.27	69.89	88.84	95.38	71.77	85.33	96.89	96.72	87.89	93.83
	CAT	42.69	68.38	80.27	63.78	85.55	94.86	81.66	87.36	96.89	97.15	88.06	94.03
	TIES	69.15	42.68	61.10	57.64	80.44	91.29	84.06	85.26	96.23	95.23	84.25	91.90
	DialectMoE	**70.58**	**78.40**	**85.08**	**78.02**	**91.24**	**96.42**	**88.73**	**92.13**	96.89	**97.99**	**88.72**	**94.53**
Gemma 2	Instruct	54.94	18.89	22.48	32.10	84.89	90.85	87.03	87.59	86.89	88.55	84.37	86.60
	LoRA Group	72.18	**81.76**	83.03	78.99	**90.80**	94.99	**89.50**	**91.76**	**96.95**	**98.40**	86.00	**93.78**
	LoRA Sep	**78.14**	78.98	**87.24**	**81.45**	89.17	**96.21**	87.83	91.07	94.53	97.99	**87.41**	93.31
	CAT	52.43	77.84	83.95	71.41	89.4	94.70	85.55	89.88	96.17	97.18	86.01	93.12
	TIES	76.08	50.56	57.10	61.25	87.59	93.23	**89.50**	90.11	95.50	94.91	87.28	92.56
	DialectMoE	46.52	69.24	79.92	65.23	88.74	91.29	82.41	87.48	83.21	89.42	76.56	83.06

Table 2: F_1 scores for zero-shot experiments across different models and techniques.

In the few-shot setting, several models (Mistral-7B Instruct, Gemma 2 with *DialectMoE*) yield lower F_1 compared to zero-shot. All models show moderate stability on AU with F_1 scores typically in the 88-90% range for Reddit and 94-99% for Google. Overall, sentiment classification tasks show strong dialect robustness, with Google sentiment demonstrating superior performance where most fine-tuned models achieve F_1 scores above 95% across dialects, while Reddit sentiment shows more variation with scores typically in the 85-96% range.

How do proposed techniques compare against baselines? In the zero-shot setting, for Reddit sentiment, Mistral-7B Instruct shows the weakest results for Indian English with F_1 score of 73.58%, whereas our proposed techniques achieve significantly higher scores. For instance, *DialectMoE* demonstrates remarkable improvements, with Mistral achieving F_1 scores above 90% for all dialects. *DialectFusion* using CAT merging in Phi-3 performs better than LoRA Grouping and Instruct with by 2-5% improvements, respectively, on the Indian English dialect. For Google Sentiment, Llama 3.1 Instruct struggles across all dialects. In contrast, *DialectMoE* shows significant improvements over the baseline ($\sim$20% each), demonstrating the effectiveness of our proposed approach. *DialectFusion* with CAT merging shows the strongest performance on Phi-3, reaching 99.19% on UK, representing the best results for this dialect. *DialectFusion* with CAT and TIES provides robust performance, with Qwen 2.5 CAT achieving 94.20% F_1 on UK and 83.69% F_1 on IN on Reddit sentiment, outperforming baselines by approximately $\sim$2%. For

Model	Technique	Reddit Sarcasm				Reddit Sentiment				Google Sentiment			
		AU	UK	IN	Avg	AU	UK	IN	Avg	AU	UK	IN	Avg
Mistral-7B	Instruct	56.07	25.88	17.79	33.25	69.36	59.61	65.69	64.89	92.55	90.29	83.33	88.72
	LoRA Grouping	58.29	63.28	61.35	60.97	**81.54**	**81.36**	73.18	78.69	**96.14**	**95.84**	83.46	**91.81**
	LoRA Separate	**63.82**	**71.71**	80.27	**71.93**	79.13	68.95	76.55	74.88	94.64	94.98	84.20	91.27
	CAT	52.92	69.74	77.67	66.78	71.93	69.58	73.10	71.54	93.90	95.08	83.98	90.99
	TIES	60.02	39.72	33.13	44.29	69.86	59.61	67.43	65.63	93.27	92.53	85.73	90.51
	DialectMoE	61.98	68.41	**82.29**	70.89	79.70	79.60	**77.45**	**78.92**	93.27	93.64	**86.07**	90.99
Phi-3	Instruct	**74.76**	50.87	59.48	61.70	84.89	91.12	83.24	86.42	94.04	96.06	84.93	91.68
	LoRA Grouping	68.84	68.22	74.68	70.58	83.60	91.71	79.76	85.02	94.73	97.20	85.05	92.33
	LoRA Separate	67.42	68.38	**80.27**	**72.02**	**87.92**	92.94	83.95	**88.27**	94.64	96.38	**88.16**	93.06
	CAT	56.07	**73.05**	79.82	69.65	84.89	93.10	84.05	87.35	**96.17**	**97.61**	86.99	**93.59**
	TIES	72.82	55.36	64.14	64.11	84.08	92.32	**84.41**	86.94	93.27	96.45	85.73	91.82
	DialectMoE	64.30	69.32	75.96	69.86	79.32	**94.51**	82.25	85.36	92.97	93.90	84.90	90.59
Qwen 2.5	Instruct	69.85	53.66	64.30	62.60	84.60	87.98	81.30	84.63	91.11	94.53	86.25	90.63
	LoRA Grouping	**71.12**	76.69	**80.97**	76.26	86.03	92.32	76.80	85.05	96.17	96.38	87.40	93.32
	LoRA Separate	70.70	**80.14**	80.27	**77.04**	**90.37**	92.54	83.30	**88.74**	**96.98**	95.95	86.47	93.13
	CAT	42.69	68.38	80.27	63.78	88.04	**94.20**	**83.69**	88.64	**96.98**	**97.58**	85.69	93.42
	TIES	65.03	70.22	78.25	71.17	83.06	88.67	80.58	84.10	94.76	96.06	89.50	**93.44**
	DialectMoE	52.29	64.33	49.62	55.41	87.42	85.63	76.19	83.08	94.73	95.58	**88.10**	92.80
Llama 3.1	Instruct	51.48	19.85	19.48	30.27	80.98	92.63	82.96	85.52	90.39	91.12	81.39	87.63
	LoRA Grouping	54.81	42.50	62.18	53.16	**89.89**	**92.48**	**86.13**	**89.50**	92.44	95.62	87.35	91.80
	LoRA Separate	61.58	68.38	80.27	70.08	81.16	82.60	73.50	79.09	88.85	94.77	82.40	88.67
	CAT	42.69	68.38	80.27	63.78	79.48	86.00	77.47	80.98	86.73	90.32	84.69	87.25
	TIES	**70.30**	43.82	49.14	54.42	78.84	90.85	85.70	85.13	91.77	95.25	84.64	90.55
	DialectMoE	53.20	**80.82**	**83.21**	**72.41**	74.56	84.65	70.26	76.49	**94.73**	**96.42**	**87.87**	**93.01**
Gemma 2	Instruct	72.35	31.48	39.09	47.64	86.85	91.43	85.19	87.82	91.87	93.73	88.83	91.48
	LoRA Grouping	57.14	**84.80**	83.21	75.05	89.64	**96.21**	84.73	90.19	93.85	**99.19**	87.40	**93.48**
	LoRA Separate	**77.49**	73.09	83.21	**77.93**	89.93	94.35	83.32	89.20	93.85	97.14	85.60	92.20
	CAT	45.45	81.61	**87.95**	71.67	**90.77**	94.86	85.63	**90.42**	**95.38**	97.98	85.60	92.99
	TIES	74.79	64.80	69.60	69.73	88.10	93.23	**87.33**	89.55	94.76	95.64	**88.57**	92.99
	DialectMoE	48.19	69.00	79.70	65.63	84.76	90.45	81.66	85.62	79.43	79.27	79.85	79.52

Table 3: F_1 scores for few-shot experiments across different models and techniques.

Google sentiment, Gemma 2 with LoRA Grouping in the few-shot setting demonstrates strong performance with F_1 reaching 99.99% for UK. *DialectMoE* performs considerably better for Llama 3.1, outperforming both baselines. *DialectFusion* with CAT provides robust performance on Gemma 2 and Qwen 2.5.

How does *DialectFusion* compare to *DialectMoE*? *DialectMoE* proves to be effective in zero-shot setting on both Reddit and Google sentiment subsets. For Reddit Sentiment, *DialectMoE* achieves remarkable performance across Llama 3.1, Qwen 2.5, and Mistral-7B, with Mistral-7B showing the strongest performance across dialects. For Google sentiment, *DialectMoE* yields high scores across multiple models, and Llama 3.1 showing significant improvements compared to its baselines.

In the few-shot setting, *DialectFusion* with CAT and TIES is more consistent and reliable. For Reddit sentiment, *DialectFusion* with CAT demonstrates strong performance compared to *DialectMoE*, with Qwen 2.5 CAT achieving 88.04% on AU, 94.20% on UK, and 83.69% on enIN with $\sim$8% advantage over the latter approach. Gemma 2 shows strong performance with both CAT and TIES. In contrast, *DialectMoE* exhibits mixed effectiveness in the few-shot scenarios. For Reddit sentiment, Mistral-7B achieves as low as 77.45% on IN, below results in the zero-shot setting. For Google sentiment, *DialectFusion* with CAT provides robust performance, with Gemma 2 and Qwen 2.5 achieving high scores. Overall, *DialectMoE* excels in zero-shot settings, while *DialectFusion* provides more stable performance across both settings.

ID	Dialect	Domain	Task	Example	True	Base	LoRA	CAT	TIES	MoE
1	UK	Reddit	Sentiment	"Looks like we got a genius over here"	0	1	1	0 (Phi)	1	0 (Llama)
2	IN	Google	Sentiment	"Very famous and renowned bhaaji box shop. Quality is good. Little overpriced."	1	0	0	0	0	1 (Llama)
3	UK	Google	Sentiment	"It's a very good pub. The food is also nice, but the price is too high for the size you get, so sorry to add this comment. I just couldn't believe 7 plus service on chicken strips. It's just not worth it eating out. I am not sure how that can be improved"	1	0	0	1 (Qwen)	0	0
4	AU	Reddit	Sentiment	"Urgent care clinics might be your next best option: It emergency but for non life threatening issues, anything life threatening I'd be calling an ambulance."	0	1	1	0 (Phi)	1	1
5	AU	Reddit	Sarcasm	"They present *opinion* from those sources. But of course, opinion that aligns with one's ideology is *analysis*. As you said, you shouldn't give credit to the news source for referring to *analysis* of others. Glad we agree those two measures quoted by The Spectator and The Guardian are inflationary (or at least I think we agreed, you haven't moved past the whole media watch thing yet)."	0	1	1	0 (Gemma)	1	0
6	AU	Reddit	Sarcasm	"The Decepticons are infiltrating and taking over"	1	1	1	0	1 (Gemma)	0
7	IN	Reddit	Sarcasm	"Was she playing pub G? On a serious note tho, om Shanti."	1	0	0	0	0	0
8	IN	Reddit	Sentiment	"bro woke up to revenge the out of context use of his speech"	1	0	0	0	0	0

Table 4: Behavior of different methods across different dialects, domains, and tasks. Base denotes the Instruct variant, LoRA denotes LoRA Grouping, CAT and TIES refer to *DialectFusion*, and MoE refers to *DialectMoE*. Examples demonstrate cases where: baselines fail but proposed methods succeed (1-4), CAT vs TIES divergence (5-6), and all methods fail (7-8).

5.3 Qualitative Analysis

Next, we provide a qualitative analysis of the behavior of baselines and our proposed methods, with examples shown in Table 4.

The first three examples show the case where the baselines fail, but our proposed methods, *DialectMoE* on Qwen 2.5 and Llama 3.1 as well as *DialectFusion* with CAT merging on Phi-3 classify the sentiment correctly. Example 2 contains dialectal vocabulary ("bhaaji") and mixed sentiment. While baselines fail, *DialectMoE* on Llama 3.1 correctly classifies the sentiment to be positive (1), suggesting its dialect-specific experts better capture regional linguistic patterns and sentiment expressions. Example 3 illustrates the peformance of *DialectFusion* with CAT in correctly identifying the overall positive sentiment, while all other methods fail. This demonstrates CAT's effective merging of dialect-specific knowledge, which appears particularly valuable for expressions in British English.

Examples 5-6 illustrate the case where CAT succeeds but TIES fails, and vice versa. Example 5 illustrates *DialectFusion* with CAT correctly identifying sarcasm distributed across multiple sentences with embedded quotes, whereas *DialectFusion* with TIES fails. On the other hand, Example 6 shows CAT failing on brief statements with clear cultural markers, whereas TIES correctly identifies this "pop culture" reference as sarcastic.

Example 2 demonstrates the case where *DialectMoE* performs correct classification while CAT and TIES fail. The reason may be that the routing mechanism successfully directs examples to specialized experts, whereas merging approaches fail to aggregate the distributed knowledge effectively when dialectal signals ("bhaaji") are strong. However, CAT demonstrates superiority when dialectal signals are ambiguous, as evident from Example 4. While *DialectMoE* fails across all models, CAT succeeds consistently.

Error Analysis We also present error cases (Examples 7 and 8) in Table 4 to reveal limitations of proposed approaches. In Example 7, all the models misclassify the example as non-sarcastic (0); this may be due to models focusing on surface level lexical cues such as "serious" and "Om Shanti", failing to capture the pop culture reference of the game called PubG which provides the sarcastic intent. Similarly, for the informal Indian English expression (Example 8) "bro woke up to revenge", all proposed methods, including baselines, fail to capture the positive sentiment towards defensive action.

Overall, the proposed approaches demonstrate promising advantages for dialect adaptation, with effectiveness varying across zero-shot and few-shot settings as well as task. For instance, *DialectMoE* performs best for both tasks in zero-shot, *Di-*

alectFusion using CAT merging is competitive on the sentiment classification task, especially on the Google subset, and LoRA Separate is especially effective in the few-shot setting where other approaches under-perform.

Although LoRA Separate achieves robust performance in most settings since each dialect has its own dedicated adapter trained on isolated dialectal examples without parameter-sharing constraints, this causes difficulties at inference time, particularly when inputs are from mixed dialects. In contrast, fusion-based approaches produce a single model, eliminating the need for adapter selection during inference and thus reducing deployment complexities and memory overhead. These methods better handle mixed or unknown dialectal inputs. While our results indicate that merging strategies do not consistently yield positive transfer, the competitive performance of CAT demonstrates an effective strategy for translating separately trained adapters into practical systems.

DialectMoE demonstrates positive transfer, particularly in zero-shot settings, indicating that the routing mechanism enables expert specialization by directing each token to the most suitable expert. The shared feed forward network captures universal English patterns, while the LoRA experts seems to specialize in dialect-specific features. Moreover, the gating mechanism can leverage multiple experts when inputs exhibit cross-dialectal characteristics. The observed performance degradation in few-shot settings may stem from overfitting due to limited few-shot examples, causing the router to focus on in-context learning as opposed to the generalization acquired during training. *DialectMoE* does not encounter difficulties when handling mixed-dialect inputs as there is a single adapter for multiple dialects where experts represent the differentiating factor. Although *DialectMoE* incurs routing computation costs, it activates only the top-k experts per token, resulting in more efficient inference time than full multi-adapter approaches, while maintaining dynamic adaptation capabilities.

6 Conclusion

In this paper, we evaluated the performance of five open-source LLMs across Australian, British, and Indian English dialects, and proposed two parameter-efficient frameworks based on LoRA adapters and mixture of experts to enhance their robustness. Our first approach, *DialectFusion* involves training separate LoRA adapters for each dialect and merging them using two techniques, CAT and TIES. Among these, CAT demonstrates stable performance improvements across most of the LLMs. Our second approach, *DialectMoE*, based on mixture-of-experts framework, shows strong overall performance, with particularly notable gains for Qwen 2.5, LLaMA 3.1, and Mistral-7B. Experimental results indicate that both approaches consistently outperform baseline methods including instruction-tuned models and standard LoRA fine-tuning, which demonstrate the effectiveness of parameter-efficient strategies for improving dialect robustness.

Limitations

Our work has several limitations. First, *DialectMoE* does not support all the models and custom implementation is needed for extending beyond the provided models which might not be a scalable approach. Additionally, our error analysis reveals that all our methods, including *DialectMoE*, struggle possibly because of their reliance on surface level lexical cues indicating limited contextual understanding. This suggests that parameter-efficient adaptation on its own might not be sufficient for capturing culture specific intent without having a complementary mechanism for external knowledge and reasoning.

Acknowledgements

Mamta and Oana Cocarascu acknowledge the support from EPSRC (grant number EP/X04162X/1).

References

arcee-ai. 2024. The-tome: A curated dataset for instruction-following language models. https://huggingface.co/datasets/arcee-ai/The-Tome. Accessed: 2025-12-10.

Fei Ding and Baiqiao Wang. 2025. Improved supervised fine-tuning for large language models to mitigate catastrophic forgetting. *arXiv e-prints*, pages arXiv–2506.

Abdin et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. Technical Report MSR-TR-2024-12, Microsoft.

Fahim Faisal and Antonios Anastasopoulos. 2024. Data-augmentation-based dialectal adaptation for LLMs. In *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*, pages 197–208, Mexico City, Mexico. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

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.

Aditya Joshi, Raj Dabre, Diptesh Kanojia, Zhuang Li, Haolan Zhan, Gholamreza Haffari, and Doris Dippold. 2025. Natural language processing for dialects of a language: A survey. *ACM Comput. Surv.*, 57(6).

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *CoRR*, abs/2404.15159.

Fangru Lin, Shaoguang Mao, Emanuele La Malfa, Valentin Hofmann, Adrian de Wynter, Xun Wang, Si-Qing Chen, Michael J. Wooldridge, Janet B. Pierrehumbert, and Furu Wei. 2025. Assessing dialect fairness and robustness of large language models in reasoning tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6317–6342, Vienna, Austria. Association for Computational Linguistics.

Yanchen Liu, William Held, and Diyi Yang. 2023. DADA: Dialect adaptation via dynamic aggregation of linguistic rules. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13776–13793, Singapore. Association for Computational Linguistics.

Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. 2025. Lora soups: Merging loras for practical skill composition tasks. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 644–655.

Dipankar Srirag, Aditya Joshi, Jordan Painter, and Diptesh Kanojia. 2025a. BESSTIE: A benchmark for sentiment and sarcasm classification for varieties of English. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8413–8429, Vienna, Austria. Association for Computational Linguistics.

Dipankar Srirag, Nihar Ranjan Sahoo, and Aditya Joshi. 2025b. Evaluating dialect robustness of language models via conversation understanding. In *Proceedings of the Second Workshop on Scaling Up Multilingual & Multi-Cultural Evaluation*, pages 24–38, Abu Dhabi. Association for Computational Linguistics.

Jiao Sun, Thibault Sellam, Elizabeth Clark, Tu Vu, Timothy Dozat, Dan Garrette, Aditya Siddhant, Jacob Eisenstein, and Sebastian Gehrmann. 2023. Dialect-robust evaluation of generated text. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6010–6028, Toronto, Canada. Association for Computational Linguistics.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Sowmya Vajjala and Shwetali Shimangaud. 2025. Text classification in the LLM era - where do we stand? *CoRR*, abs/2502.11830.

Gido M. van de Ven, Nicholas Soures, and Dhireesha Kudithipudi. 2025. *Continual learning and catastrophic forgetting*, page 153–168. Elsevier.

Sajana Weerawardhena, Paul Kassianik, Blaine Nelson, Baturay Saglam, Anu Vellore, Aman Priyanshu, Supriti Vijay, Massimo Aufiero, Arthur Goldblatt, Fraser Burch, Ed Li, Jianliang He, Dhruv Kedia, Kojin Oshiba, Zhouran Yang, Yaron Singer, and Amin Karbasi. 2025. Llama-3.1-foundationai-securityllm-8b-instruct technical report. *CoRR*, abs/2508.01059.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, and 9 others. 2025. Qwen2.5-1m technical report. *CoRR*, abs/2501.15383.

Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2025. Sarcasmbench: Towards evaluating large language models on sarcasm understanding. *IEEE Transactions on Affective Computing*, 16(4):2560–2578.

Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta, and Diyi Yang. 2023. Multi-VALUE: A framework for cross-dialectal English NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 744–768, Toronto, Canada. Association for Computational Linguistics.

A Implementation details

We implemented the proposed approaches using the MoE-PEFT library[4] which is an LLMOps framework designed for high-throughput finetuning, evaluation and inference. We modified the

[4]https://github.com/TUDB-Labs/MoE-PEFT

original implementations to account for prompt templates of various instruct models so the fine-tuning remains consistent. In case of DialectMoE, to represent number of dialects (en-AU, en-UK, en-IN), we set the number of experts to 3 for every LLM except Mistral. We assign the number of experts to 6 for sarcasm classification and 8 for sentiment classification because the lower number of experts did not result in convergence. Similarly, the learning rates were lowered to 1e-5 from 2e-4 for convergence (DialectMoE).

Since our experiments focus on instruction-tuned LLMs, it was important to prevent catastrophic forgetting of their instruction-following capabilities while fine-tuning them on the BESSTIE dataset (van de Ven et al., 2025). For this reason, we augment the BESSTIE dataset with FineTome-100k. The FineTome-100k dataset is a filtered version of The Tome dataset, which consisted of high quality multi-turn conversation compiled from 9 public datasets and curated using a reranker, educational value scoring and composite scoring (arcee-ai, 2024). We augment all our training datasets with 2000 samples of FineTome-100k (Ding and Wang, 2025) and also transformed each sample into a user-assistant message pair with task-specific prompts as mentioned below. Finally, we also validate that all conversations maintain proper role alternation i.e. starting with user, ending with assistant and no consecutive messages from the same role. This helps in filtering out any malformed example and also perform de-duplication.

A.1 Prompt Templates

We use two task-specific prompts as outlined in the BESSTIE paper, one for each task. For sentiment classification, we then prompt the model with: *"Generate the sentiment of the given text. 1 for positive sentiment, and 0 for negative sentiment. Do not give an explanation."*

Similarly, for sarcasm classification, we use: *"Predict if the given text is sarcastic. 1 if the text is sarcastic, and 0 if the text is not sarcastic. Do not give an explanation."*

Given the text and a task-specific prompt, the expected behavior of the LLM is to generate either 1 (for positive) or 0 (for negative).