

Where Do Predictions Go? Class-Level Shifts and Carbon Cost in Stance Detection

Nityaa Kalra, Jisk Attema and Dimitar Shterionov

Tilburg University

{N.Kalra, J.J.Attema, D.Shterionov}@tilburguniversity.edu

Abstract

Stance detection is widely used to summarize public discourse by measuring support, opposition, and neutrality towards societal issues. Evaluation pipelines rely on aggregate metrics such as macro-F1 that can mask how models actually distribute predictions across stance classes. We study this limitation by comparing large language and (statistical) supervised models across three stance detection datasets with different definitions of neutrality. Our evaluation considers predictive performance, class-level behavior, and environmental impact. We find that zero/ few-shot language models do not consistently outperform supervised baselines, and some models exhibit class-level priors that persist across datasets and prompting strategies. In addition, performance gains tend to saturate in the 4B–8B parameter range. These findings suggest that aggregate performance scores alone are insufficient for responsible stance detection. When model outputs are used to measure public opinion at scale, class-level behavior and deployment cost matter as much as the overall accuracy.

1 Introduction

Stance detection is the task of identifying whether a given text expresses support, opposition, or no clear position towards a given target. In domains such as health, climate change, education, and politics, it is used as a proxy for public opinion and societal response to issues. Because of this, the way models assign stance labels can affect how researchers interpret patterns of agreement, disagreement, or uncertainty.

Following the advances and rapid adoption of large language models (LLMs) for many NLP tasks, recent work has also shown the applicability of LLMs on the task of stance detection. However, comparisons with traditional supervised models often rely on closed proprietary models, with only a few open-model exceptions (Zhang et al., 2022;

Zhao et al., 2024; Cruickshank and Ng, 2026). This limits reproducibility and makes practical trade-offs between models harder to assess. In addition, most evaluations focus on aggregate scores over the *favor* and *against* classes, often excluding the neutral class.

We argue that the neutral class should be explicitly considered in the evaluation, as its meaning varies across datasets. For example, SemEval-2016 (Mohammad et al., 2016) defines *none* as the absence of a stance, whereas EZ-STANCE (Zhao and Caragea, 2024) includes topically related examples that lack sufficient evidence for support or opposition. Since stance predictions are often aggregated to summarize public discourse, such variation can change how agreement, disagreement, and neutrality are represented. Therefore, evaluating models across all three stance categories is necessary for a more complete assessment of their behavior.

In this work, we evaluate LLMs for stance detection across three (and not two) stance classes. We also address the question of model selection, which involves trade-offs between performance, model size, environmental cost, and whether a model is open or closed. To study these trade-offs, we ask:

- **RQ1:** How do language models differ from supervised baselines in their distribution of stance predictions across datasets with different definitions of stance classes?
- **RQ2:** How small can language models be before performance degrades meaningfully, and is the switch to language models worth their additional computational or environmental cost?

To support reproducibility of our work, we release all code in our GitHub¹ repository.

¹<https://github.com/nityaak5/slm-topic-modeling>

2 Literature Review

2.1 Stance Detection

Stance detection identifies whether a text expresses support, opposition, or no clear position towards a given target. Unlike sentiment analysis, it models the relationship between a text and a specific target rather than overall sentiment. SemEval-2016 (Mohammad et al., 2016) established a standard benchmark for the task, while subsequent datasets extended it to zero-shot topics (Allaway and McKeown, 2020) and more challenging neutral cases (Zhao and Caragea, 2024). Target-aware models have improved performance by explicitly modeling the relationship between text and target (Augenstein et al., 2016; Du et al., 2017), although cross-target stance detection remains challenging (Allaway and McKeown, 2020; Jamadi Khiabani and Zubiaga, 2025). Further, evaluation also varies across datasets. SemEval-style setups typically report macro-F1 over *favor* and *against*, excluding the neutral class (Mohammad et al., 2016), whereas datasets such as EZ-STANCE emphasize challenging neutral examples (Zhao and Caragea, 2024). These differences motivate our class-level analysis alongside aggregate metrics.

2.2 LLMs for Stance Detection

Recent work has explored the use of LLMs for stance detection using zero-shot, few-shot, and reasoning-based prompting (Cruickshank and Ng, 2026; Jamadi Khiabani and Zubiaga, 2025; Ma et al., 2024; Gül et al., 2024; Taranukhin et al., 2024; Liyanage et al., 2024). While some studies report improvements from reasoning-based prompting (Zhang et al., 2025), others find that performance depends on the dataset, prompt design, and model family (Liyanage et al., 2024; Cruickshank and Ng, 2026). Fine-tuning also does not consistently outperform prompting-based approaches (Cruickshank and Ng, 2026). In addition, the widespread use of closed LLMs raises concerns about data contamination and model bias (Aiyappa et al., 2023; Ng et al., 2024). These limitations motivate comparisons between different types of language models under controlled evaluation settings.

2.3 Impact of Size on Model Performance

Scaling studies show that increasing model parameter count generally improves performance across NLP tasks (Kaplan et al., 2020; Hoffmann

et al., 2022). However, the benefits of scaling vary across tasks, and smaller models fine-tuned on task-specific data can often match or outperform larger models (Schick and Schütze, 2021). Compute-optimal scaling further suggests that improving training efficiency can be more beneficial than increasing parameter count alone (Hoffmann et al., 2022). These findings motivate model selection based on the trade-off between predictive performance and computational requirements. Since stance detection is often applied to large collections of social and political text, larger models also incur higher computational and carbon costs (Faiz et al., 2023; Cruickshank and Ng, 2026). We therefore evaluate models based on both predictive performance and carbon footprint, in line with Green AI principles (Schwartz et al., 2019).

3 Methodology

3.1 Datasets and Models

We evaluate on three English stance detection datasets that differ in source, domain, and class distribution. SemEval-2016 (Mohammad et al., 2016) contains tweet-level annotations with *favor*, *against*, and *none* labels. EZ-STANCE (Zhao and Caragea, 2024) is a zero-shot dataset with noun-phrase and claim targets; we use only the noun-phrase subset (EZStance-NP), and MT²-CSD (Niu et al., 2025) contains Reddit conversations. We exclude the *Post-T* target for consistency. These datasets differ in their class balance: SemEval is skewed towards *against*, EZStance-NP is relatively balanced, and MT²-CSD contains more neutral cases. Dataset statistics are shown in Table 1. Here, targets denote specific issues or entities towards which a stance is expressed, while domains represent broader topical categories.

Dataset	Year	Test Size	Coverage
SemEval-2016	2016	1249	5 targets
EZStance-NP	2024	2663	8 domains
MT ² -CSD	2025	2371	5 targets

Table 1: Summary of the datasets.

We evaluate closed and open-weight LLMs across multiple model families and scales. Closed models include GPT-5.2, GPT-5-mini, and Claude-Haiku-4.5, while open-weight models cover Qwen, Llama, DeepSeek, Nemotron, and Phi families (1.7B–14B parameters; details in Appendix G).

3.2 Prompting and Inference Setup

Following (Cruickshank and Ng, 2026), we evaluate models using 11 prompting strategies (see Appendix A), from zero-shot to chain-of-thought and 15-shot in-context learning. All prompts include the input text, target, task description, label space, and require a single stance label as output.

Few-shot examples are sampled randomly from each training set using a fixed seed with class-balanced sampling. Open models are served via vLLM (Kwon et al., 2023) on an NVIDIA H100 NVL GPU cluster, while closed models are accessed via APIs. Temperature is set to 0 where supported.

3.3 Evaluation Metrics

We report Macro F1 over *favor*, *against*, and *none* as the primary metric. Macro F1 is defined as:

$$\text{Macro F1} = \frac{1}{3} \sum_{c \in C} F1_c, \quad (1)$$

where $C = \{\text{favor}, \text{against}, \text{none}\}$ and $F1_c$ is the class-wise F1 score. For MT²-CSD, we report only 2-class F1 due to the lack of a reproducible implementation of the baseline. 3-class Macro F1 results of other models are available on our GitHub. We report 95% bootstrap confidence intervals for all models (Appendix F).

To analyze class-level behavior, we compute *prediction shift* for each class, defined as the difference between its predicted share and gold-label share. Positive values indicate over-prediction, while negative values indicate under-prediction.

We also estimate environmental impact using CO₂ equivalent emissions. For open models, we use CarbonTracker (Anthony et al., 2020), and for closed models we rely on EcoLogits (Rincé and Banse, 2025). Since hardware and energy mix are not fully observable for API-accessible models, these measurements are just estimates. To approximate deployment-scale cost, we extrapolate average per-instance emissions to one million inputs:

$$\bar{E}_{1M} = \bar{e} \times 10^6, \quad (2)$$

where $\bar{e}$ is the mean per-instance CO₂eq across prompts and instances.

4 Results and Discussion

4.1 Class-level Analysis

Consistent with prior work (Cruickshank and Ng, 2026), macro-F1 varies across datasets and model

families (Appendix E). On SemEval-2016, several closed models outperform the SVM baseline, with Phi-4 as the strongest open model. On EZStance-NP, models outperform the zero-shot ChatGPT baseline but not the supervised baseline, while on MT²-CSD all models except Qwen3-1.7B outperform the supervised baseline in the two-class setting. Further, more elaborate prompting or including additional in-context examples does not consistently improve model performance. These results motivate a closer examination of how predictions are distributed across stance classes.

On SemEval-2016, most models under-predict *against*, shifting predictions mainly to *none* (mean: +12.7 pp) and, to a lesser extent, *favor* (mean: +6.9 pp) (Figure 1). The confusion matrices (Appendix B) confirm this for 11 of 14 models. On MT²-CSD, both *against*→*none* and *favor*→*none* hold consistently across 13 of 14 models regardless of family, with *favor*→*none* reaching as high as 58%. In contrast, EZStance-NP consistently shifts predictions from *none* to *favor* for all models, likely because neutral examples are made more challenging by removing target mentions. Across all datasets, Llama-3.1-8B and Nemotron-4B consistently under-predict *none*, indicating a stable preference for directional stance labels.

Overall, **even though language models achieve competitive aggregate performance, their class-level predictions are inconsistent across datasets. The shifts we observe are dataset- and model-dependent, and largely unaffected by prompting strategy alone.**

4.2 Model Scaling and Carbon Cost

We next examine how model scaling affects performance and carbon cost. Across the Qwen3 family (Figure 2), performance improves sharply from 1.7B to 4B parameters (e.g., 0.393 to 0.625 on SemEval-2016), but gains beyond 4B are marginal (e.g., 0.625 to 0.630 from 4B to 14B), with overlapping bootstrap confidence intervals confirming these differences are not statistically reliable (Appendix F). Although absolute performance differs across model families, **scaling beyond the 4B–8B range provides limited performance gains.**

Carbon cost follows a different trend. Within the same family, smaller models are generally more efficient (e.g., Qwen3-8B is 14–27% cheaper than Qwen3-14B with comparable performance), but parameter count is not a reliable predictor across families. For example, Phi-4 (14B) is more carbon

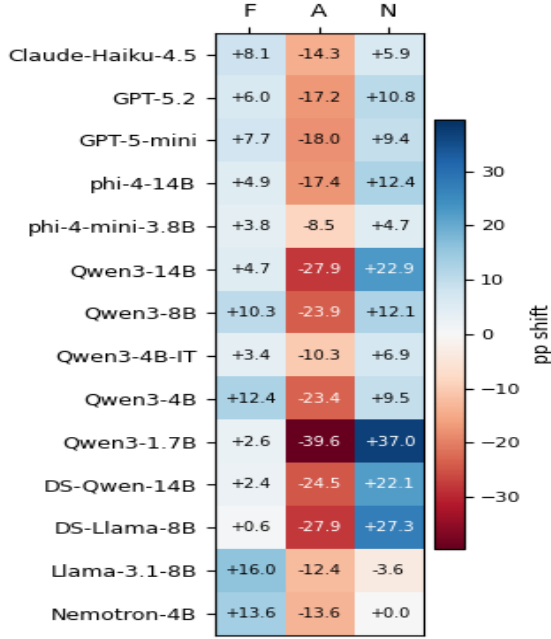


Figure 1: Prediction shift (in percentage points (pp)) on SemEval-2016, averaged across all prompting strategies. F = Favor, A = Against, N = None. Figures for EZStance-NP and MT²-CSD are in Appendix C

efficient than Qwen3-4B, while Phi-4-mini (3.8B) and Nemotron-4B are more efficient than Qwen3-1.7B despite their larger size. Among closed models, Claude-Haiku-4.5 is the most carbon-efficient overall, while GPT-5-mini is the most expensive model, likely due to longer outputs under chain-of-thought prompting. These results suggest that **architecture and inference behavior have a greater influence on carbon cost than model size alone**.

This being said, supervised models are generally more computationally and environmentally efficient than language models, which makes the case for adopting language models slightly weaker. Therefore, LLMs should be preferred only when they provide a clear advantage over supervised alternatives along the accuracy–cost frontier.

5 Conclusion

In this work, we studied stance detection models from a class-level perspective, examining how prediction behavior, model scale, and environmental cost vary across model families. Across three datasets, models with similar macro-F1 scores often produce different distributions of *favor*, *against*, and *none*, revealing behaviors that remain hidden when evaluation focuses only on aggregate performance.

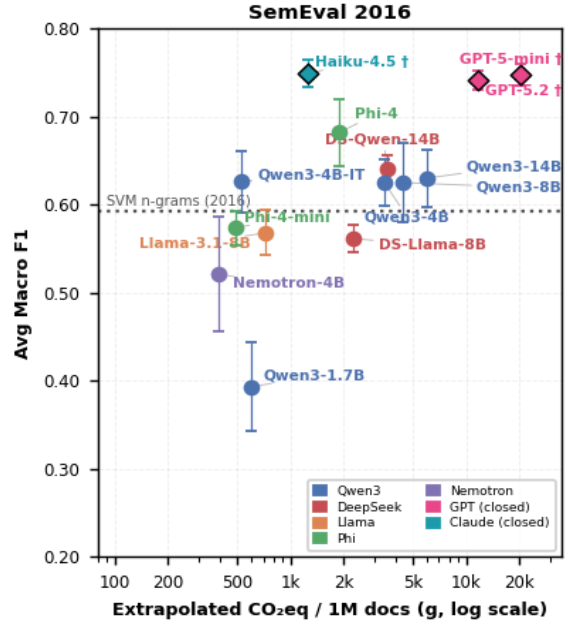


Figure 2: Performance vs extrapolated carbon cost on SemEval-2016. Error bars show deviation in Macro F1 across prompting strategies. Corresponding figures for EZStance-NP and MT²-CSD are in Appendix D

Performance gains for open LLMs saturate around the 4B–8B parameter range, while carbon cost is influenced more by architecture than parameter count. Closed models generally achieve stronger performance, although the gap varies across datasets. Among them, Claude-Haiku-4.5 offers a strong performance–carbon trade-off, showing that efficient deployment does not necessarily require the largest or most expensive models.

These findings raise an important deployment question. Closed models may offer strong performance but limited transparency, while open models provide greater accessibility but can exhibit persistent biases towards directional classes regardless of dataset or prompt. In both cases, aggregate metrics alone are insufficient to reveal these differences. If the same model becomes the default across research and monitoring pipelines, its biases risk becoming the standard and may shape how public opinion is measured rather than truly reflecting it.

Finally, supervised models remain more efficient at deployment scale, suggesting that LLMs should be preferred only when they provide clear gains on the accuracy–cost frontier.

Future work should extend this analysis to more languages, domains, and target-level evaluations under class imbalance. Comparing model predictions with human annotations could help distin-

guish model errors from annotation ambiguity or instruction effects. Evaluating finer-grained efficiency measures, such as latency and energy variability, alongside datasets less likely to overlap with pretraining data would further support deployment-aware evaluation.

6 Limitations

This study has several limitations. First, we evaluate three datasets in English with three stance labels covering various domains. These data, however, do not represent the full range of stance detection settings such as sarcasm, multilingual data, and culturally specific forms of political expression. Additionally, SemEval-2016 is a decade old dataset and is likely to be present in the training data of some of the language models evaluated here, which may partly explain stronger performance on this dataset.

Second, in MT²-CSD, we classify individual conversational lines without full dialog context, meaning that a *none* prediction may sometimes reflect missing context rather than the absence of stance. Our setup also excludes the *Post-T* target to maintain comparability across datasets. This removes part of the conversational structure that may be important for stance interpretation.

Third, our results depend on the prompts, parsing rules, and model versions we used. We did not experiment with fine-tuned models or assess whether the observed prediction shifts persist after calibration (Zhao et al., 2021).

Our carbon analysis focuses on inference-time emissions. Open-model emissions are measured on our hardware, while closed-model emissions are estimates. The one-million-document analysis assumes linear scaling and does not account for factors such as batching, quantization, or hardware utilization. These factors matter in real deployments and should be included in future.

Finally, our class-level redistribution analysis shows how model predictions differ from gold distributions, but it does not establish downstream impact. Future work should examine the effects of over- or under-predicting particular stance classes depending on the application, e.g. media monitoring or public-health communication.

7 Ethical and Societal Implications

Stance detection is often used in real-world applications such as media monitoring or climate policy analysis. In these settings, model outputs are often

aggregated to summarize how population responds to the issues. It means that differences in model behavior accumulate across large numbers of predictions and shape how agreement, disagreement, and uncertainty are represented in downstream analysis. A model that consistently avoids neutrality or shifts opposition into other categories produces a biased view of the underlying discourse that may not be visible from aggregate evaluation alone.

This raises practical concerns. Closed models often achieve better performance, but are less accessible under budget constraints and provide limited visibility into their prediction behavior. This makes it difficult to identify or correct systematic biases. Open models are more accessible and transparent, but, as we show, carry stable model-level priors, which persist across datasets and prompting strategies. Neither option is bias-free, and without class-level evaluation, these differences remain invisible to practitioners making deployment decisions.

Environmental cost adds another dimension to these trade-offs. We find that carbon cost is driven more by model architecture than parameter count, and that several mid-sized open models achieve competitive performance at substantially lower cost than larger or closed alternatives.

The positive contribution of this work lies in making these trade-offs visible. By moving beyond aggregate metrics and examining how models distribute predictions across stance classes, we provide a more complete view of model behavior and environmental cost before deployment. We hope this encourages class-level evaluation and carbon-aware model selection as standard practices, especially in applications where stance outputs inform decisions about public issues.

References

- Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yongyeol Ahn. 2023. [Can we trust the evaluation on ChatGPT?](#) In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 47–54, Toronto, Canada. 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 (EMNLP)*, pages 8913–8931, Online. Association for Computational Linguistics.

Lasse F. Wolff Anthony, Benjamin Kanding, and

- Raghavendra Selvan. 2020. [Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models](#). *arXiv preprint*.
- Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. [Stance Detection with Bidirectional Conditional Encoding](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.
- Iain J. Cruickshank and Lynnette Hui Xian Ng. 2026. [Prompting and Fine-Tuning Open Source Large Language Models for Stance Classification](#). *ACM Transactions on Intelligent Systems and Technology*, 17(4):1–25.
- Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. [Stance Classification with Target-specific Neural Attention](#). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3988–3994, Melbourne, Australia. International Joint Conferences on Artificial Intelligence Organization.
- Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Osi, Prateek Sharma, Fan Chen, and Lei Jiang. 2023. [LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models](#).
- İlker Gül, Rémi Lebret, and Karl Aberer. 2024. [Stance Detection on Social Media with Fine-Tuned Large Language Models](#). *arXiv preprint*. ArXiv:2404.12171.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego De Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George Van Den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, and 3 others. 2022. [Training Compute-Optimal Large Language Models](#). In *Advances in Neural Information Processing Systems 35*, pages 30016–30030, New Orleans, Louisiana, USA. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Parisa Jamadi Khiabani and Arkaitz Zubiaga. 2025. [Cross-target stance detection: A survey of techniques, datasets, and challenges](#). *Expert Systems with Applications*, 283:127790.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling Laws for Neural Language Models](#). *arXiv preprint*. ArXiv:2001.08361.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient Memory Management for Large Language Model Serving with PagedAttention](#). *arXiv preprint*.
- Chandreen R. Liyanage, Ravi Gokani, and Vijay Mago. 2024. [GPT-4 as an X data annotator: Unraveling its performance on a stance classification task](#). *PLOS ONE*, 19(8):e0307741.
- Junxia Ma, Changjiang Wang, Hanwen Xing, Dongming Zhao, and Yazhou Zhang. 2024. [Chain of Stance: Stance Detection with Large Language Models](#). *arXiv preprint*. ArXiv:2408.04649.
- Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. [SemEval-2016 Task 6: Detecting Stance in Tweets](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.
- Lynnette Hui Xian Ng, Iain Cruickshank, and Roy Ka-Wei Lee. 2024. [Examining the Influence of Political Bias on Large Language Model Performance in Stance Classification](#). *arXiv preprint*. ArXiv:2407.17688.
- Fuqiang Niu, Genan Dai, Yisha Lu, Jiayu Liao, Xiang Li, Hu Huang, and Bowen Zhang. 2025. [MT2-CSD: A New Dataset and Multi-Semantic Knowledge Fusion Method for Conversational Stance Detection](#). *arXiv preprint*.
- Samuel Rincé and Adrien Banse. 2025. [EcoLogits: Evaluating the Environmental Impacts of Generative AI](#). *Journal of Open Source Software*, 10(111):7471.
- Timo Schick and Hinrich Schütze. 2021. [Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.
- Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. [Green AI](#). *arXiv preprint*.
- Maksym Taranukhin, Vered Shwartz, and Evangelos Milios. 2024. [Stance Reasoner: Zero-Shot Stance Detection on Social Media with Explicit Reasoning](#). *arXiv preprint*.
- Bowen Zhang, Daijun Ding, Liwen Jing, Genan Dai, and Nan Yin. 2022. [How would Stance Detection Techniques Evolve after the Launch of ChatGPT?](#) *arXiv preprint*.
- ZhaoDan Zhang, Jin Zhang, Hui Xu, Jiafeng Guo, and Xueqi Cheng. 2025. [MPRF: Interpretable Stance Detection through Multi-Path Reasoning Framework](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 454–470, Suzhou, China. Association for Computational Linguistics.
- Chenye Zhao and Cornelia Caragea. 2024. [EZ-STANCE: A Large Dataset for English Zero-Shot](#)

[Stance Detection](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15697–15714, Bangkok, Thailand. Association for Computational Linguistics.

Chenye Zhao, Yingjie Li, Cornelia Caragea, and Yue Zhang. 2024. [ZeroStance: Leveraging ChatGPT for Open-Domain Stance Detection via Dataset Generation](#). In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13390–13405, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. [Calibrate before use: Improving few-shot performance of language models](#). In *International conference on machine learning*, pages 12697–12706. Pmlr.

A Prompting Strategies

No-def.

Your task is to determine the stance of the following document toward the given query.

QUERY: [QUERY]

DOCUMENT: [DOCUMENT]

Return valid JSON in exactly this format: {"stance": "LABEL"}

The "stance" value must be exactly one of: "FAVOR", "AGAINST", "NONE".

Default.

Your task is to determine the stance of the following document toward the given query.

QUERY: [QUERY]

DOCUMENT: [DOCUMENT]

Return valid JSON in exactly this format: {"stance": "LABEL"}

The "stance" value must be exactly one of: "FAVOR", "AGAINST", "NONE".

Use "FAVOR" when the document overall supports the query, "AGAINST" when it overall opposes the query, and "NONE" when it discusses the query without clearly supporting or opposing it.

Task-def.

Stance classification is the task of determining the expressed or implied opinion, or stance, of a document toward a certain, specified target. Analyze the following document and determine its stance toward the provided query.

QUERY: [QUERY]

DOCUMENT: [DOCUMENT]

Return valid JSON in exactly this format: {"stance": "LABEL"}

The "stance" value must be exactly one of: "FAVOR", "AGAINST", "NONE".

Use "FAVOR" only when the author is definitely in favor of the query. Use "AGAINST" only when the author is definitely against the query. Use "NONE" if any of the following holds: (a) the document does not discuss the query at all, (b) the document discusses it but the author takes no clear side (neutral/balanced), or (c) the author's position cannot be determined with confidence. Do not guess from indirect hints.

Scale.

Stance classification is the task of determining the expressed or implied opinion, or stance, of a document toward a certain, specified target. Analyze the following document and rate its stance toward the query on this scale:

1 = strongly against

2 = somewhat against

3 = neither for nor against (neutral)

4 = somewhat in favor

5 = strongly in favor

query: [QUERY]

document: [DOCUMENT]

Return ONLY valid JSON in exactly this format:

{"stance_score": <integer between 1 and 5>}

The "stance_score" value must be an integer and exactly one of: 1, 2, 3, 4, 5.

This score will be mapped as: 1/2 -> "AGAINST", 3 -> "NONE", 4/5 -> "FAVOR".

Do not return any text outside the JSON object.

Question.

Stance classification is the task of determining the expressed or implied opinion, or stance, of a statement toward a certain, specified target. What is the stance of the following social

media statement toward the following entity?

entity: [QUERY]

statement: [DOCUMENT]

Return ONLY valid JSON in exactly this format: {"stance": "LABEL"}

The "stance" value must be exactly one of: "FAVOR", "AGAINST", "NONE".

Use "FAVOR" when the statement supports the entity, "AGAINST" when it opposes the entity, and "NONE" when it is neither clearly supporting nor opposing.

CoT (two-step).

Step 1 (elicit reasoning):

Stance classification is the task of determining the expressed or implied opinion, or stance, of a statement toward a certain, specified target. Think step-by-step and explain whether the social media statement supports, opposes, or is neither for the query.

query: [QUERY]

document: [DOCUMENT]

Return ONLY valid JSON in exactly this format:

{"reasoning": "your explanation"}

Step 2 (final label, conditioned on Step 1 output [REASONING]):

Therefore, based on your explanation, [REASONING], what is the final stance? Return ONLY valid JSON in exactly this format: {"stance": "LABEL"}.

The "stance" value must be exactly one of: "FAVOR", "AGAINST", "NONE". Use "FAVOR" when the statement supports the entity, "AGAINST" when it opposes the entity, and "NONE" when it is neither clearly supporting nor opposing.

entity: [QUERY]

statement: [DOCUMENT]

stance:

3/6/9/12/15-shot.

Shared template; only the number of in-context examples $N \in \{3, 6, 9, 12, 15\}$ changes.

Stance classification is the task of determining the expressed or implied opinion, or stance, of a document toward a specified query.

Use exactly one label: "FAVOR", "AGAINST", or "NONE".

Use "FAVOR" when the document supports the query, "AGAINST" when it opposes the query, and "NONE" when it is neither clearly supporting nor opposing.

Examples:

query: [EXAMPLE QUERY]

document: [EXAMPLE DOCUMENT]

{"stance": "[EXAMPLE STANCE]"}

... (repeated N times, $N/3$ per class) ...

Now classify the following item.

query: [QUERY]

document: [DOCUMENT]

Return ONLY valid JSON in exactly this format: {"stance": "LABEL"}

The "stance" value must be exactly one of: "FAVOR", "AGAINST", "NONE".

B Confusion Matrices per Model

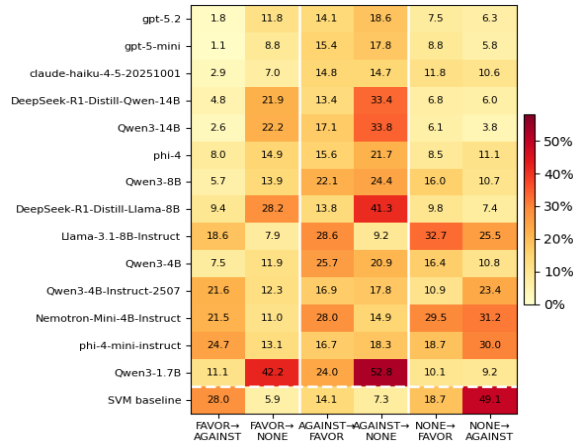


Figure 3: Confusion matrix (mean across prompting strategies) on SemEval-2016.

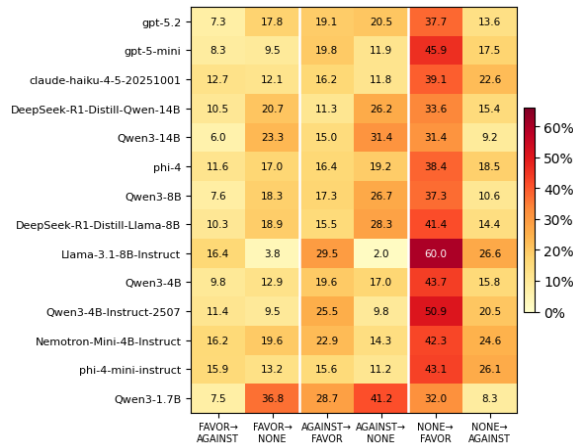


Figure 4: Confusion matrix (mean across prompting strategies) on EZStance-NP.

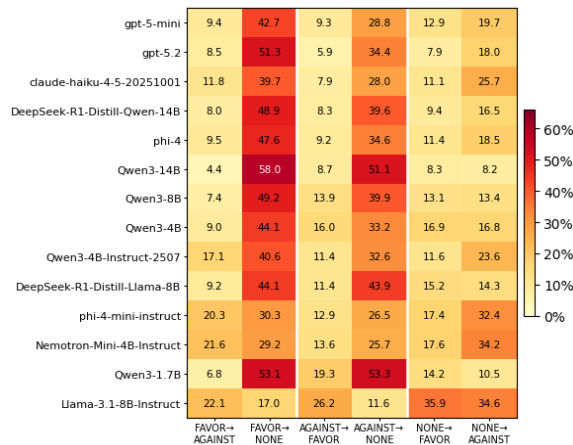


Figure 5: Confusion matrix (mean across prompting strategies) on MT²-CSD.

C Class-level Prediction Shift

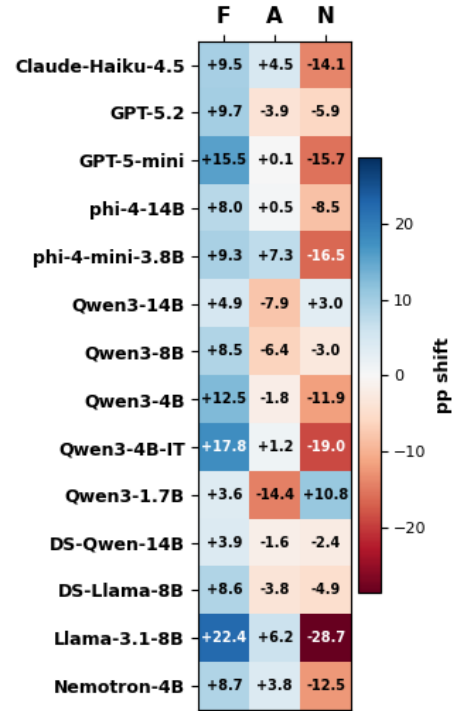


Figure 6: Class-level prediction shift (in percentage points (pp)) on EZStance-NP, averaged across all prompting strategies. F = Favor, A = Against, N = None.

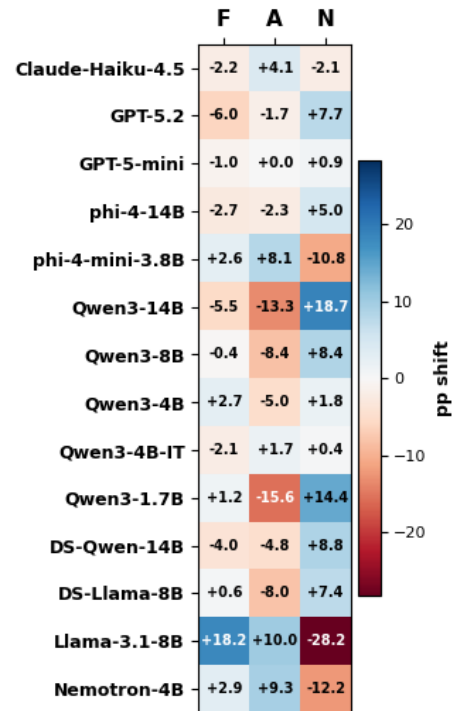


Figure 7: Class-level prediction shift (in percentage points (pp)) on MT²-CSD averaged across all prompting strategies. F = Favor, A = Against, N = None.

D Carbon Cost-Performance Analysis

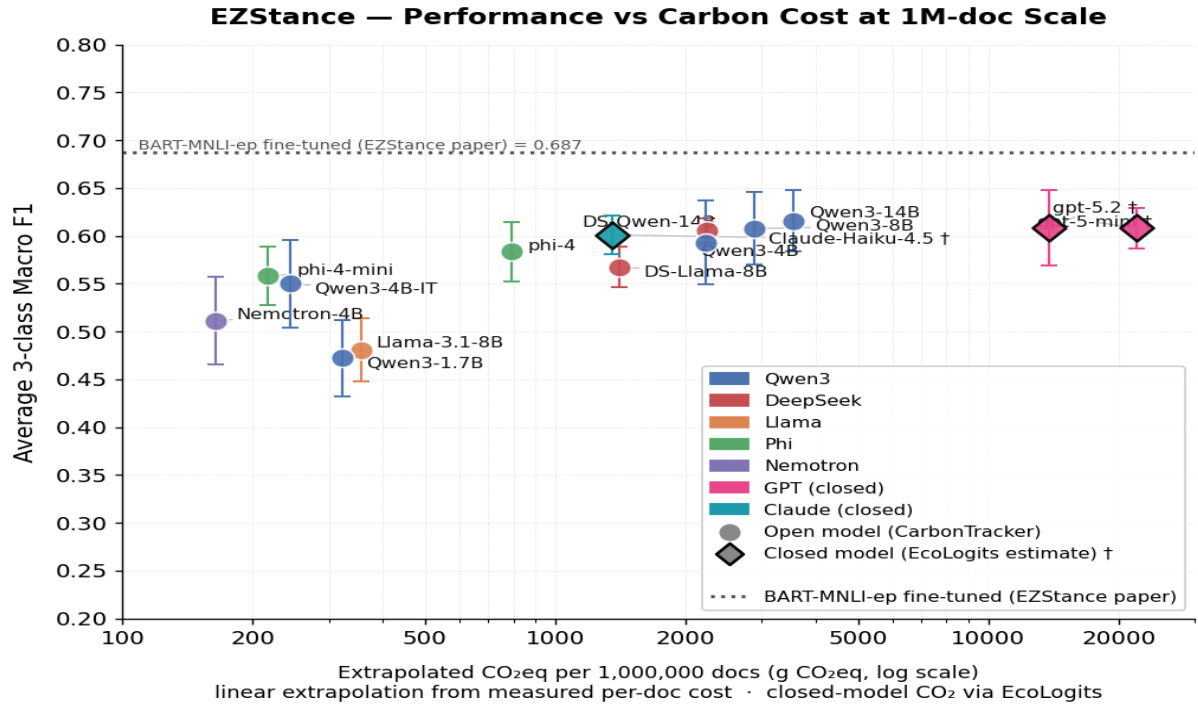


Figure 8: Performance vs extrapolated carbon cost on EZStance-NP. Error bars show deviation in Macro F1 across prompting strategies.

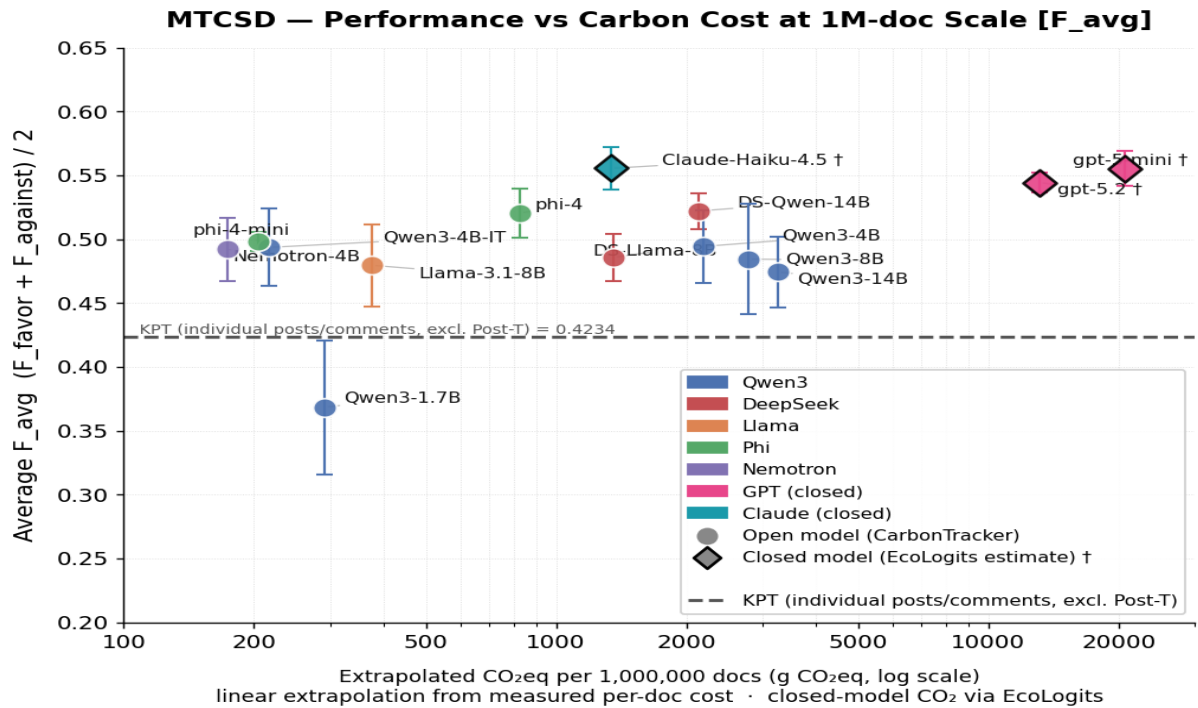


Figure 9: Performance vs extrapolated carbon cost on MT²-CSD. Error bars show deviation in F1 score across prompting strategies. We report only 2-class results due to the lack of a reproducible implementation for the baseline. 3-class Macro F1 scores are available on Github.

E Macro-F1 across Datasets

Model	No-def	Default	Task-def	Question	CoT	Scale	3-shot	6-shot	9-shot	12-shot	15-shot
Qwen3-1.7B	0.370	0.347	0.290	0.363	0.473	0.415	0.349	0.324	0.373	0.425	0.401
Qwen3-4B	0.596	0.589	0.607	0.654	0.606	0.628	0.657	0.662	0.649	0.670	0.643
Qwen3-4B-IT	0.601	0.582	0.599	0.636	0.639	0.561	0.651	0.674	0.676	0.668	0.674
Qwen3-8B	0.578	0.529	0.581	0.617	0.577	0.618	0.674	0.668	0.673	<u>0.696</u>	0.678
Qwen3-14B	0.586	0.559	0.569	0.628	0.619	0.637	0.639	0.657	0.632	<u>0.668</u>	0.658
DS-Llama-8B	0.585	0.546	0.537	0.567	0.574	0.594	0.546	0.548	0.561	0.550	0.552
DS-Qwen-14B	<u>0.623</u>	<u>0.612</u>	0.606	0.641	<u>0.652</u>	<u>0.653</u>	0.627	0.640	0.636	0.655	0.639
Llama-3.1-8B	<u>0.582</u>	<u>0.609</u>	0.582	0.593	<u>0.585</u>	<u>0.529</u>	0.515	0.578	0.528	0.569	0.561
Phi-4-mini	0.591	0.556	0.563	0.572	0.574	0.539	0.597	0.602	0.601	0.566	0.566
Phi-4	0.599	0.593	<u>0.648</u>	<u>0.672</u>	0.690	0.718	<u>0.689</u>	<u>0.702</u>	<u>0.675</u>	0.687	<u>0.695</u>
Nemotron-4B	0.503	0.512	<u>0.466</u>	<u>0.483</u>	0.496	0.425	<u>0.545</u>	<u>0.565</u>	<u>0.632</u>	0.652	<u>0.622</u>
GPT-5-mini	0.745	0.737	0.744	0.723	0.723	0.767	0.736	0.740	0.756	0.755	0.748
GPT-5.2	0.728	0.737	0.746	0.749	0.741	0.769	0.739	0.758	0.744	0.745	0.742
Haiku-4.5	0.715	0.754	0.765	0.766	0.740	0.766	0.748	0.749	0.745	0.751	0.730

Table 2: SemEval-2016 Task 6 (3-class Macro-F1). Baseline = 0.592 (SVM n-grams). **Bold** = best per column; underline = best open-source model per column.

Model	No-def	Default	Task-def	Question	CoT	Scale	3-shot	6-shot	9-shot	12-shot	15-shot
Qwen3-1.7B	0.443	0.438	0.434	0.461	0.533	0.500	0.418	0.447	0.484	0.522	0.515
Qwen3-4B	0.477	0.560	0.579	0.601	0.599	0.637	0.603	0.610	0.608	0.630	0.621
Qwen3-4B-IT	0.529	0.464	0.545	0.563	0.596	0.472	0.542	0.580	0.585	0.582	0.593
Qwen3-8B	0.518	0.564	0.580	0.623	0.608	0.633	<u>0.622</u>	0.635	0.632	0.633	0.635
Qwen3-14B	0.536	0.579	0.607	0.642	<u>0.630</u>	0.644	0.620	0.627	0.628	0.634	0.625
DS-Llama-8B	0.540	0.525	0.555	0.573	0.569	0.560	0.582	0.578	0.584	0.579	0.598
DS-Qwen-14B	0.587	<u>0.591</u>	<u>0.622</u>	0.623	0.597	0.609	0.595	0.599	0.618	0.609	0.611
Llama-3.1-8B	0.514	0.465	0.489	0.496	0.498	0.503	0.408	0.436	0.488	0.474	0.514
Phi-4	0.554	0.565	0.587	0.567	0.604	0.512	0.594	0.602	0.606	0.611	0.615
Phi-4-mini	0.521	0.499	0.553	0.537	0.559	0.566	0.551	0.574	0.585	0.595	0.597
Nemotron-4B	0.502	0.531	0.506	0.497	0.462	0.402	0.531	0.548	0.547	0.549	0.547
GPT-5-mini	0.579	0.590	0.630	0.632	0.572	0.591	0.606	0.621	0.620	0.626	0.621
GPT-5.2	0.516	0.584	0.601	0.640	0.632	0.570	0.626	0.630	0.625	0.632	0.629
Haiku-4.5	0.573	0.603	0.604	0.602	0.633	0.563	0.586	0.609	0.610	0.615	0.611

Table 3: EZStance-NP (3-class Macro-F1). Baseline = 0.669 (BART-MNLI-ep, M→N). **Bold** = best per column; underline = best open-source model.

Model	No-def	Default	Task-def	Question	CoT	Scale	3-shot	6-shot	9-shot	12-shot	15-shot
Qwen3-1.7B	0.330	0.313	0.321	0.369	0.459	0.461	0.339	0.344	0.386	0.398	0.332
Qwen3-4B	0.443	0.443	0.484	0.506	0.493	0.483	<u>0.518</u>	0.519	0.514	0.510	0.523
Qwen3-4B-IT	0.473	0.439	0.501	0.515	0.493	0.439	<u>0.506</u>	0.507	0.523	0.518	0.515
Qwen3-8B	0.416	0.409	0.452	0.485	0.465	0.505	0.497	0.522	0.522	0.526	<u>0.529</u>
Qwen3-14B	0.444	0.423	0.465	0.499	0.459	0.490	0.466	0.462	0.502	0.507	0.502
DS-Llama-8B	0.488	0.451	0.466	0.489	0.509	0.513	0.469	0.491	0.492	0.494	0.482
DS-Qwen-14B	<u>0.524</u>	0.502	0.513	0.536	<u>0.532</u>	<u>0.546</u>	0.502	0.512	<u>0.532</u>	0.526	0.516
Llama-3.1-8B	0.507	<u>0.511</u>	0.514	0.518	0.499	0.479	0.450	0.453	0.423	0.457	0.464
Phi-4	0.501	<u>0.475</u>	<u>0.521</u>	<u>0.539</u>	0.515	0.538	0.515	0.522	0.526	<u>0.542</u>	0.526
Phi-4-mini	0.498	0.484	0.506	0.499	0.498	0.503	0.498	0.511	0.491	0.500	0.490
Nemotron-4B	0.508	0.483	0.462	0.491	0.493	0.435	0.506	<u>0.525</u>	0.505	0.500	0.504
GPT-5-mini	0.544	0.543	0.557	0.556	0.523	0.558	0.564	0.563	0.568	0.571	0.561
GPT-5.2	0.538	0.531	0.543	0.544	0.535	0.549	0.542	0.550	0.547	0.557	0.551
Haiku-4.5	0.548	0.531	0.542	0.548	0.530	0.557	0.567	0.568	0.578	0.570	0.572

Table 4: MT²-CSD (2-class F_{avg}). Baseline = 0.423 (KPT, individual posts). **Bold** = best per column; underline = best open-source model. We report only 2-class results due to the lack of a reproducible implementation for the baseline. 3-class Macro-F1 results of other models are available on GitHub.

F Bootstrap Confidence Intervals for Model Performance

Figures below report 95% bootstrap confidence intervals (1,000 replicates). For zero-shot prompts, intervals reflect test-set sampling variance; the point estimate is the mean across six strategies. For few-shot prompts, each replicate additionally draws a seed from 42, 101, 202, capturing exemplar-selection variance; the point estimate is the mean across three seeds. Overlapping intervals, as seen for Qwen3-4B through Qwen3-14B, indicate differences not statistically distinguishable from noise.

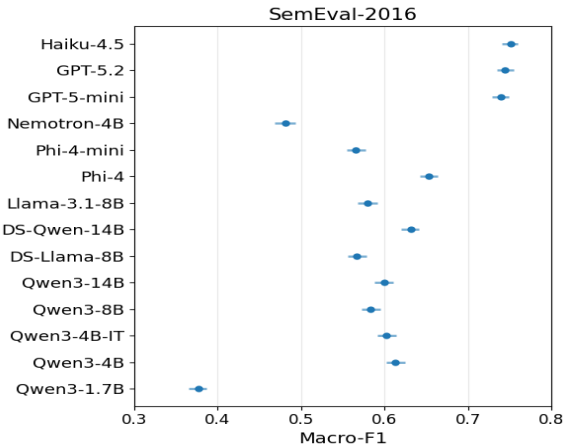


Figure 10: Macro-F1 with 95% bootstrap confidence intervals on SemEval-2016, averaged across zero-shot prompting strategies.

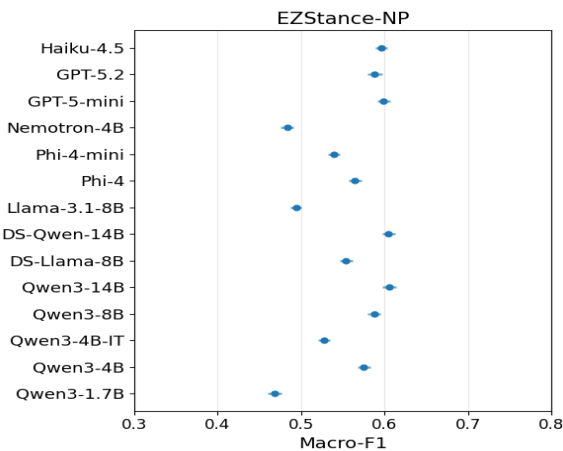


Figure 11: Macro-F1 with 95% bootstrap confidence intervals on EZStance-NP, averaged across zero-shot prompting strategies.

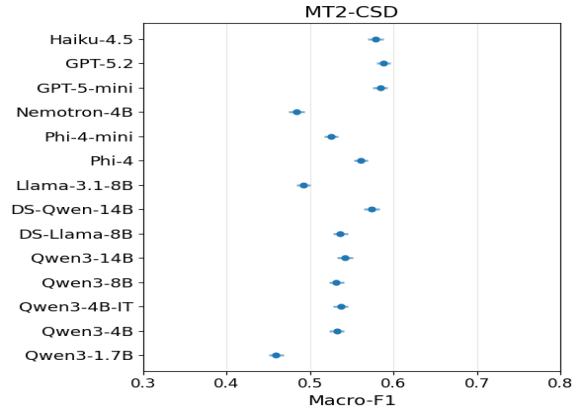


Figure 12: Macro-F1 with 95% bootstrap confidence intervals on MT²-CSD, averaged across zero-shot prompting strategies.

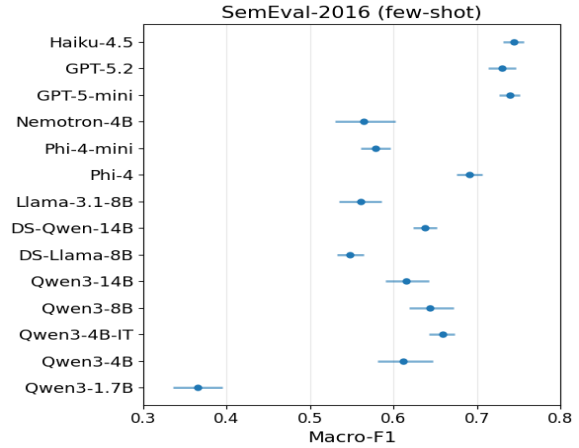


Figure 13: Macro-F1 with 95% bootstrap confidence intervals on SemEval-2016, averaged across few-shot prompting strategies. Intervals reflect selection variance across seeds {42, 101, 202}

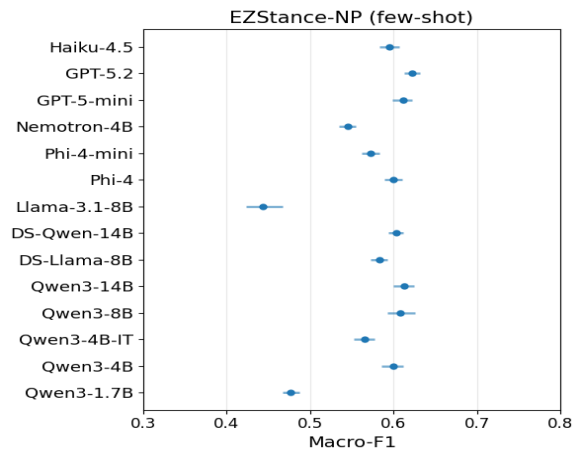


Figure 14: Macro-F1 with 95% bootstrap confidence intervals on EZStance-NP, averaged across few-shot prompting strategies. Intervals reflect selection variance across seeds {42, 101, 202}

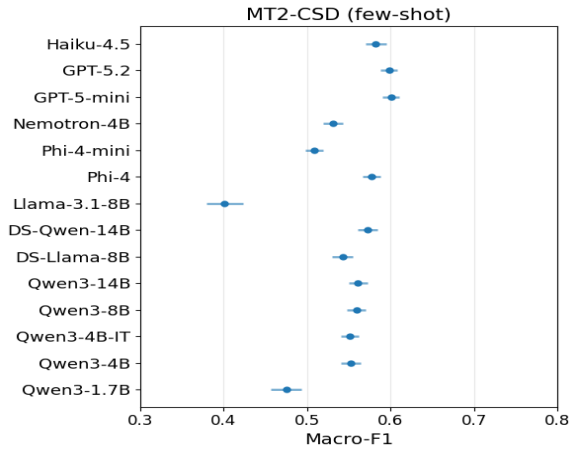


Figure 15: Macro-F1 with 95% bootstrap confidence intervals on MT²-CSD (3-class), averaged across few-shot prompting strategies. Intervals reflect selection variance across seeds {42, 101, 202}

G Model Details

Model	Provider	Params
Qwen3-1.7B	Qwen	1.7B
Qwen3-4B	Qwen	4B
Qwen3-4B-IT-2507	Qwen	4B (instruct)
Qwen3-8B	Qwen	8B
Qwen3-14B	Qwen	14B
DeepSeek-R1-Llama-8B	DeepSeek	8B (distilled)
DeepSeek-R1-Qwen-14B	DeepSeek	14B (distilled)
Llama-3.1-8B-IT	Meta	8B (instruct)
phi-4	Microsoft	14B
phi-4-mini-IT	Microsoft	3.8B (instruct)
Nemotron-4B-IT	NVIDIA	4B (instruct)
Claude-Haiku-4.5 (2025)	Anthropic	closed
GPT-5-mini (2025)	OpenAI	closed
GPT-5.2 (2025)	OpenAI	closed

Table 5: Models used in this study.