Towards Automated Rumour Stance Classification

Anonymous Author

1 Introduction

False rumours on social media could significantly impact important political outcomes [1]. Fact-checking is largely manual and struggles to scale with the quantity of rumours, so automated rumour verification is an important research area. RumourEval 2017 [5] provides a benchmark for this effort.

RumourEval Subtask A (SDQC classification) requires models to classify the stance of a tweet towards a rumour into one of four categories: [15; 22; 3]:

- **Support**: supports the veracity of the rumour.
- **Deny**: denies the veracity of the rumour.
- **Query**: asks for additional evidence regarding the veracity of the rumour.
- **Comment**: does not have a clear stance towards the veracity of the rumour (e.g. background or meta-commentary).

This goal is to automatically track reactions to a rumour, which is key for prioritising which rumours to spend time verifying [15].

In the following, we evaluate fine-tuned BERTweet [13] and prompting Llama-3.2-3b-Instruct [6] for this subtask.

2 Dataset

The RumourEval 2017 Subtask A dataset [3] contains several twitter threads, each with a source and multiple replies, as in Figure 1. To analyse it, we remove emojis, mentions, URLs and topic names (e.g. 'Sydney siege') which are not informative for stance.

The official dataset split is heavily imbalanced, with 60-75% of data labelled as comment (Table 1). Moreover, support is much rarer in the test data than other sets, which may impact performance for that class.

Table 2 summarises the top unigrams and bigrams for reply tweets broken down by SDQC label. Most notably, we see that 'lie' tokens signal denial

SDQC support classification. Example 1:

u1: We understand there are two gunmen and up to a dozen hostages inside the cafe under siege at Sydney.. ISIS flags remain on display #7News **[support]**

u2: @u1 not ISIS flags **[deny]**

u3: @u1 sorry - how do you know it's an ISIS flag? Can you actually confirm that? **[query]**

u4: @u3 no she can't cos it's actually not **[deny]**

u5: @u1 More on situation at Martin Place in Sydney, AU –LINK– **[comment]**

u6: @u1 Have you actually confirmed its an ISIS flag or are you talking shit **[query]**

SDQC support classification. Example 2:

u1: These are not timid colours; soldiers back guarding Tomb of Unknown Soldier after today's shooting #StandforCanada –PICTURE– **[support]**

u2: @u1 Apparently a hoax. Best to take Tweet down. **[deny]**

u3: @u1 This photo was taken this morning, before the shooting. **[deny]**

u4: @u1 I don't believe there are soldiers guarding this area right now. **[deny]**

u5: @u4 wondered as well. I've reached out to someone who would know just to confirm that. Hopefully get response soon. **[comment]**

u4: @u5 ok, thanks. **[comment]**

Figure 1: Two example threads from the dataset, labelled with the stance for classification. Each thread contains one source and both direct and nested replies. Figure from [3].

Table 1: Class proportions in train, dev, and test sets.

Stance	Train (%)	Dev (%)	Test (%)
comment	64.51	61.57	74.17
support	19.84	24.56	8.96
deny	7.86	3.91	6.77
query	7.79	9.96	10.10

and 'source' tokens signal query tweets (for example, they might ask for a source).

Figure 2 shows that sources are much more likely to have a stance than not, which is expected; most sources are news services and labelled support. As such, the tokens that are characteristic of source tweets with a stance are event-oriented and factual.

We perform Latent Dirichlet Allocation (LDA) [14] topic modelling, which assumes that documents can be represented as a mixture of topics, each of which is a probability distribution over tokens. We used a coherence model [17] to determine the minimum number of human-interpretable LDA topics, which was 6.

Table 3 and Figure 3 show that stance topics are very event-oriented with words like 'police', 'isis', and 'hostage'. Comment topics, however, are much more emotional and opinionated, with words like 'sad', 'safe', and 'lol' appearing often. There is some topic overlap ('kill' appears in 3 stance topics), suggesting that some tokens may be commonly used across stances. A larger number of topics would reduce the overlap, but it would be harder to visualise.

Table 2: Top 10 unigrams and bigrams in reply tweets broken down by tweet stance label. Notable findings are in bold.

Stance	Top Unigrams	Top Bigrams
Support	hostage, police, paris, break, kill, cafe, say, report, news, gunman	paris photo, supermarket paris, hostage free, free jewish, jewish supermarket, photo thomas, thomas samson, samson afp, french alps, airbus a320
Deny	flag, say, know, like, report, **lie**, news, police, hostage, need	isis flag, need help, islamic flag, parliament hill, stop spread, hostage hold, hostage situation, shahada flag, ray hadley, mike brown
Query	police, report, hostage, **source**, right, tell, know, think, kill, people	mike brown, escape cafe, look like, isis flag, hostage escape, police confirm, pay driver, michael brown, police chief, tomorrow night
Comment	police, people, like, know, kill, hostage, think, say, shoot, need	parliament hill, look like, war memorial, fire parliament, sound like, stay safe, 2009 police, police beat, beat man, man charge

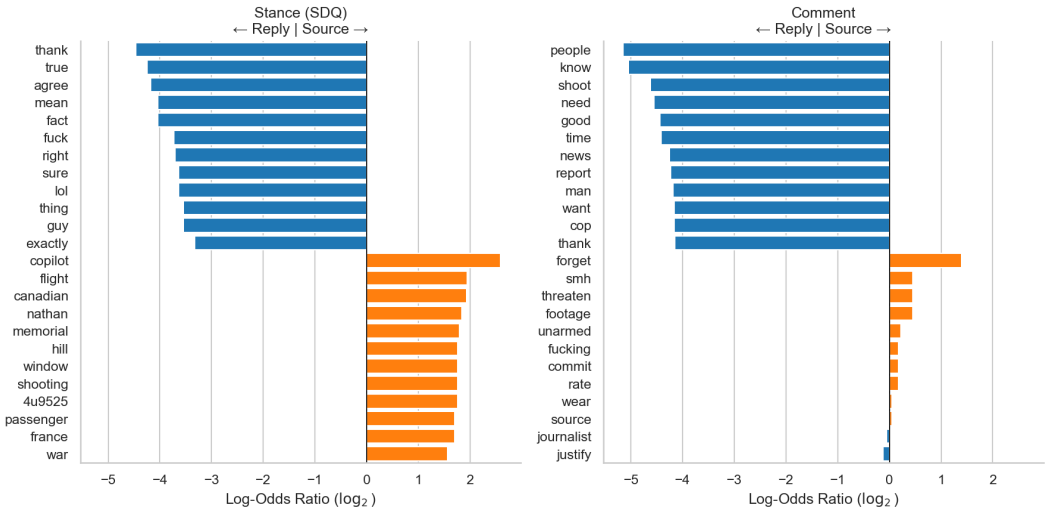

Figure 2: Top 12 tokens for stance tweets (left) and comment tweets (right), with log-odds ratios to compare token distributions between source (orange) and reply (blue) text. Positive values indicate tokens more characteristic of source tweets, while negative values indicate tokens more characteristic of reply tweets.

Table 3: LDA topic word lists for stance (SDQ) topics and comment topics, where $\texttt{n\_topics} = 6$ and the top 10 words per topic are shown.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
Stance Topics						
1	right	flag	police	hostage	report	paris
2	police	hostage	hostage	hold	news	hostage
3	suspect	isis	say	think	break	police
4	need	know	kill	look	attack	kill
5	kill	islamic	hour	people	video	break
6	know	news	situation	cafe	source	photo
7	shoot	take	cafe	like	lie	supermarket
8	confirm	report	black	report	happen	tell
9	look	gunman	people	mike	hear	afp
10	help	people	good	want	shoot	french
Comment Topics						
1	like	know	police	parliament	sad	thank
2	safe	people	want	police	right	break
3	look	go	kill	report	time	paris
4	hope	try	man	hill	happen	news
5	soldier	lol	black	say	life	need
6	send	pay	canada	people	religion	hostage
7	say	white	help	gun	2009	family
8	trump	dead	good	fire	law	crash
9	shoot	shoot	hostage	think	exactly	police
10	isis	think	need	shot	family	take

Figure 3: Word clouds of LDA topic distributions for stance (SDQ) and comment replies. Shows top 20 terms per topic.

3 Classification

3.1 Fine-Tuned BERTweet

Table 4: Model and training configuration.

Parameter	Value
Base model	`BERTweet-large`
Max input tokens	256
LoRA rank (r)	20
LoRA alpha (α)	48
LoRA dropout	0.1
LoRA target modules	query, value
Learning rate	4×10^{-5}
Optimiser	`AdamW`
Batch size	8
Weight decay	0.05
LR scheduler	Linear with warm-up
Warm-up ratio	0.15
Loss function	Weighted cross-entropy
SDQC class weights (2 d.p.)	(1.26, 3.18, 3.21, 0.39)
Mixed precision	fp16 (CUDA only)

We fine-tune BERTweet [13] with LoRA [7] (using `peft` [12])for 4-way classification, using the setup in Table 4. These hyperparameters were selected after a thorough sweep using a Bayesian search with a Gaussian Process Regression surrogate model [16] via Weights & Biases. BERTweet has high performance on tweet-related tasks like POS-tagging, NER, and text classification, and the pretrained tokenizer effectively handles emojis, URLs, user mentions and abbreviations that standard BERT-based models [4; 11] do not.

We use weighted cross-entropy loss to mitigate data imbalance. The SDQC class weights are defined:

$$w_i = \frac{N}{C \cdot n_i}$$

where w_i is the weight assigned to class $i \in \{1, \ldots, 4\}$, N is the total number of samples in the dataset, $C = 4$ is the number of classes, and n_i is the number of samples belonging to class i. This inverse frequency weighting penalises the model more heavily for misclassifying minority stance classes (SDQ) compared to the majority class (C). We trialled focal loss [10] for more sophisticated imbalance mitigation, but it performed worse in our experiments.

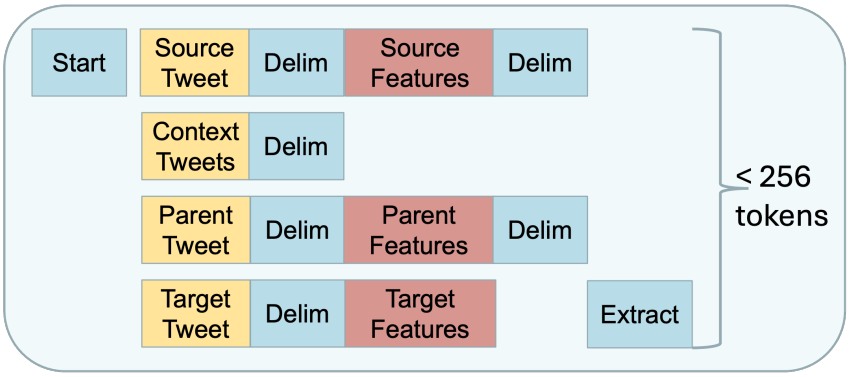

Figure 4: Overview of classifier input format, as in [21].

Our preprocessing reflects BERTweet's original training: we normalise tweets by converting emojis to text, and URLs and user mentions to special tokens `HTTPURL` and `@USER` respectively [13]. We also perform task-specific preprocessing, like previous successful RumourEval participants [8; 21]: we extract specific features such as punctuation types (!, ?) and number of swear words and negation words from tweets, and provide them as input to the model. Moreover, context has proven to be essential for stance classification [5; 3]. Therefore, as in [21], we concatenate the input as shown in Figure 4. We include the source tweet, target tweet, direct parent tweet of the target (if applicable), and any other 'context' tweets connecting the source and parent. This especially improves performance when a target tweet denies its parent, which in turn denies the source. Such a target may actually support the source, but the model needs the context in-between to learn that. We limit input to 256 tokens and truncate if necessary, starting with the context tweets to preserve the source/parent/target tweets which are most important for performance. BERTweet-large can handle up to 512 tokens, but our experiments showed no performance improvement and training took 3x longer compared to our 256 token limit.

Figure 5 shows the training diagnostics using the described setup. The validation loss increases after epoch 7 while training loss decreases, which suggests overfitting. However, the validation macro-F1 continues to increase, suggesting that the model continues to improve on minority stance classification.

3.1.1 Test Performance

Table 5 and Figure 6 show that the model effectively classifies comment (0.77 F1) and query (0.73 F1) instances, but struggles with support (0.34 F1) and deny (0.35 F1) instances. Since the training data is 65% comment, the model was expected to tend to default to comment. However, strong performance in query classification and surprisingly competitive recall for support and deny suggest that the weighted loss and input features (which

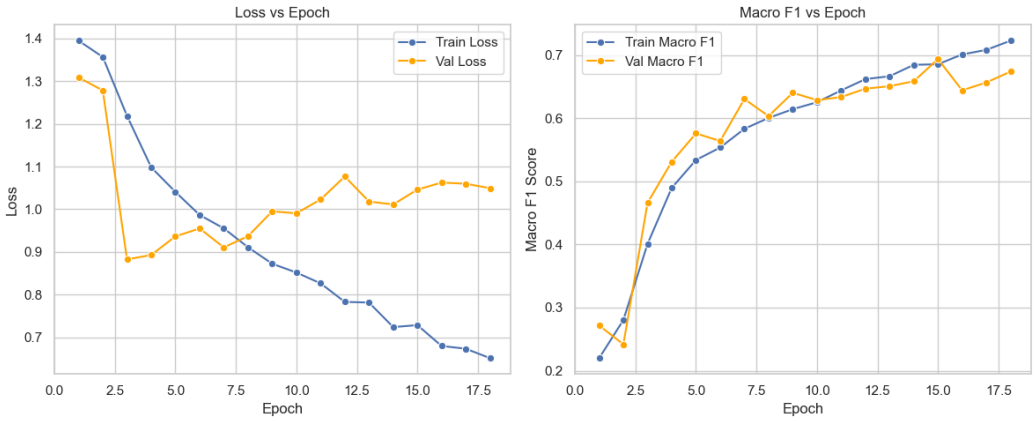

Figure 5: Training diagnostics from the run with the highest validation macro-F1, using the described setup.

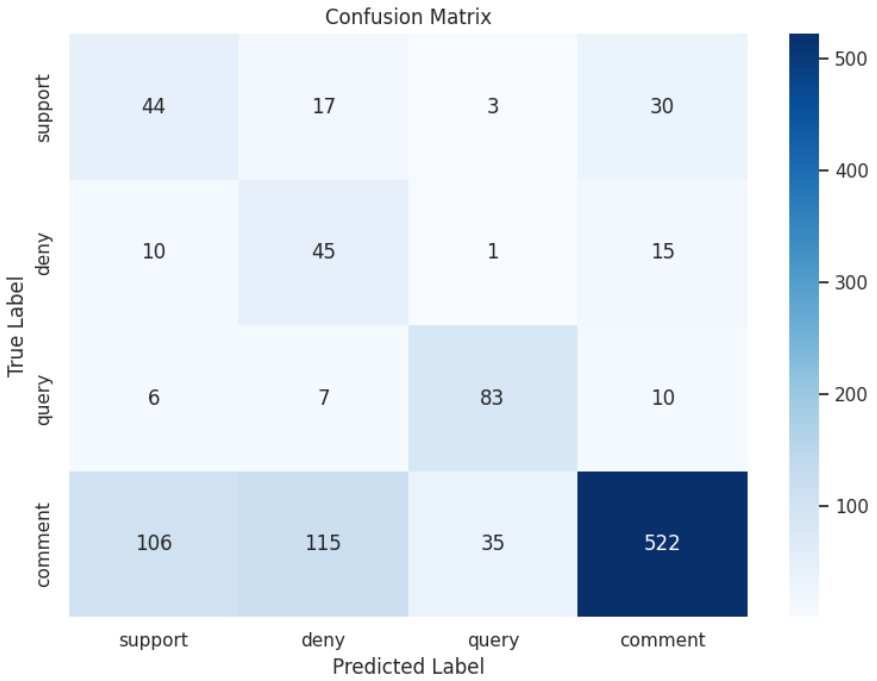

Figure 6: Confusion matrix on the test set for the fine-tuned BERTweet parameters with the highest validation macro-F1.

Table 5: Test classification performance for the fine-tuned BERTweet parameters with the highest validation macro-F1.

Stance	Precision	Recall	F1	Support
support	0.27	0.47	0.34	94
deny	0.24	0.63	0.35	71
query	0.68	0.78	0.73	106
comment	0.90	0.67	0.77	778
accuracy			0.66	1049
macro avg	0.52	0.64	0.55	1049
weighted avg	0.78	0.66	0.70	1049

includes the number of question marks) somewhat mitigated this behaviour. However, the model is prone to misclassifying instances as support and deny, as evidenced by the low precision for both stances. These high false-positive rates, combined with the higher-than-expected recalls and Figure 6 (most incorrect support/deny classifications are comment instances), suggests that the model too aggressively classifies comment instances as support and deny, perhaps because of the large penalty attached to misclassifying a minority class. Overall, the weak performance on support and deny instances is a significant barrier to automated rumour verification, as these stances are perhaps most informative for determining rumour veracity.

3.2 Basic Prompting

We use Llama-3.2-3b-Instruct [6] through Transformers [20] because it is lightweight, fast, and performs well on industry benchmarks for small models. Its knowledge cut-off is December 2023, which is good since our dataset is from 2017. We also tried 8b models, which consistently performed 5-10% better but inference was very slow. Our model choice, however, enables fast iteration which is more useful for our goal of comparing prompting methods (rather than maximising performance).

We trial zero-shot and few-shot prompting, using the system prompt and examples in Figure 7. We ablate the system prompt (Figure 8) and find it almost doubles the zero-shot macro-F1. We also use Outlines [19] to force the model to choose one of the SDQC stances. Otherwise, the model often hallucinated labels such as ‘Retweet’ and ‘Repost’ when the target tweet contained ‘RT’. To compare performance against the BERTweet baseline, we provide the model with the same input format (Figure 4) without the context tweets: these slowed inference and did not significantly increase prompting performance.

We additionally test 3 strategies for few-shot example selection:

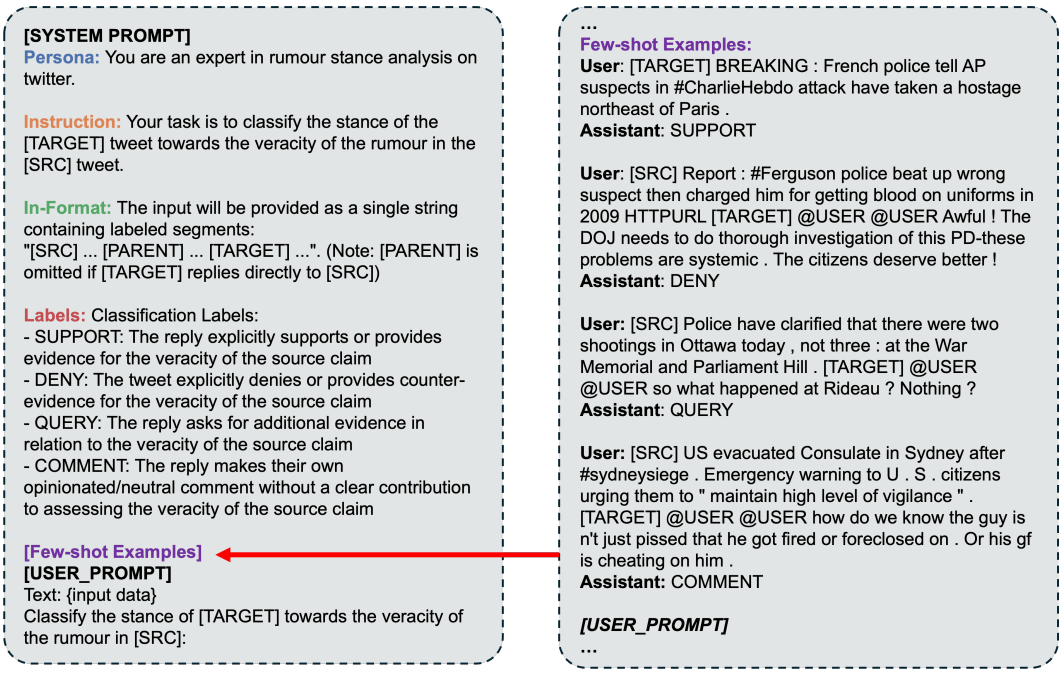

Figure 7: Exact prompt template with the few-shot examples used in our experiments. These examples were selected from the training data according to the 'diverse' strategy.

- 'Diverse': select from different topics (and source tweets).
- 'Same_source': select from the same source tweet.
- 'Random': select randomly.

Figure 9 shows the diverse strategy is best, so we use it for classification on the test set.

3.2.1 Performance

Table 6: Side-by-side comparison of classification performance on the test set for basic prompting. P = precision, R = recall.

(a) Zero-shot report.

Stance	P	R	F1	Support
support	0.23	0.03	0.06	94
deny	0.12	0.68	0.21	71
query	0.15	0.91	0.26	106
comment	0.89	0.01	0.02	778
accuracy			0.15	1049
macro avg	0.35	0.41	0.14	1049
weighted avg	0.70	0.15	0.06	1049

(b) Few-shot report.

Stance	P	R	F1	Support
support	0.16	0.60	0.26	94
deny	0.17	0.28	0.21	71
query	0.24	0.82	0.37	106
comment	0.85	0.24	0.37	778
accuracy			0.33	1049
macro avg	0.35	0.48	0.30	1049
weighted avg	0.68	0.33	0.35	1049

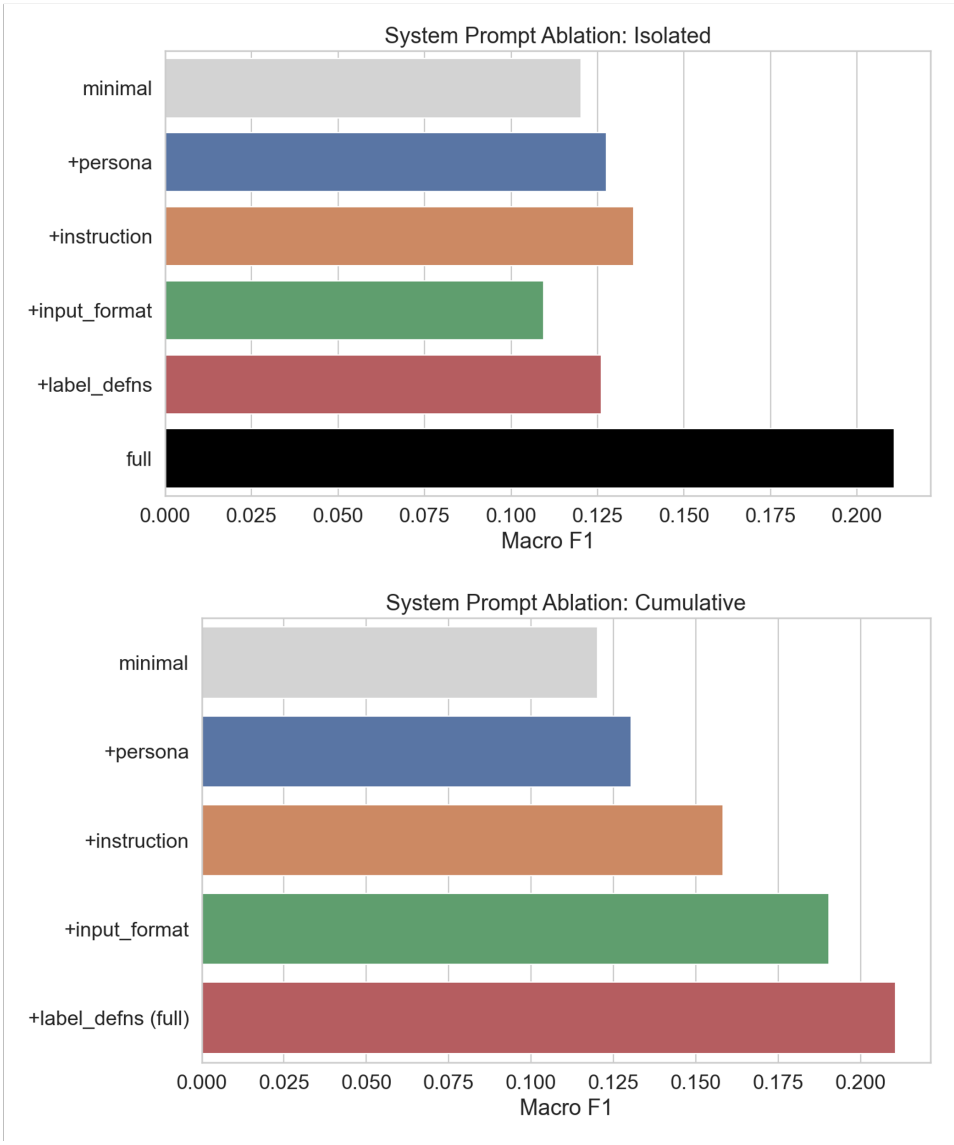

Figure 8: System prompt ablations using validation macro-F1 score. Minimal = no system prompt (just user prompt). Top: individual contribution of each system prompt component towards zero-shot macro-F1 score. Bottom: cumulative contribution to macro-F1 as each component is added to the system prompt.

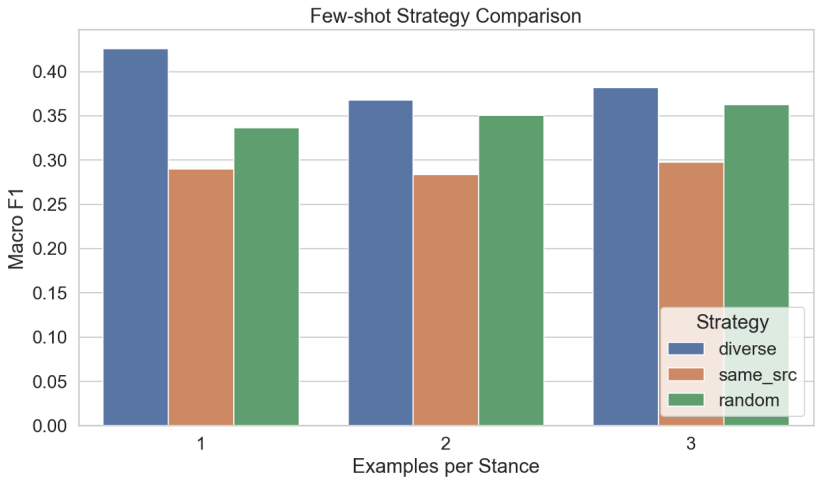

Figure 9: Comparison of three few-shot example selection strategies using validation macro-F1 score. In each case, we select the same number of examples from each stance, so the x-axis corresponds to the number of examples for each stance class. (e.g. x=1 means 4 examples, one for each stance. x=2 means 8 examples, 2 for each stance.)

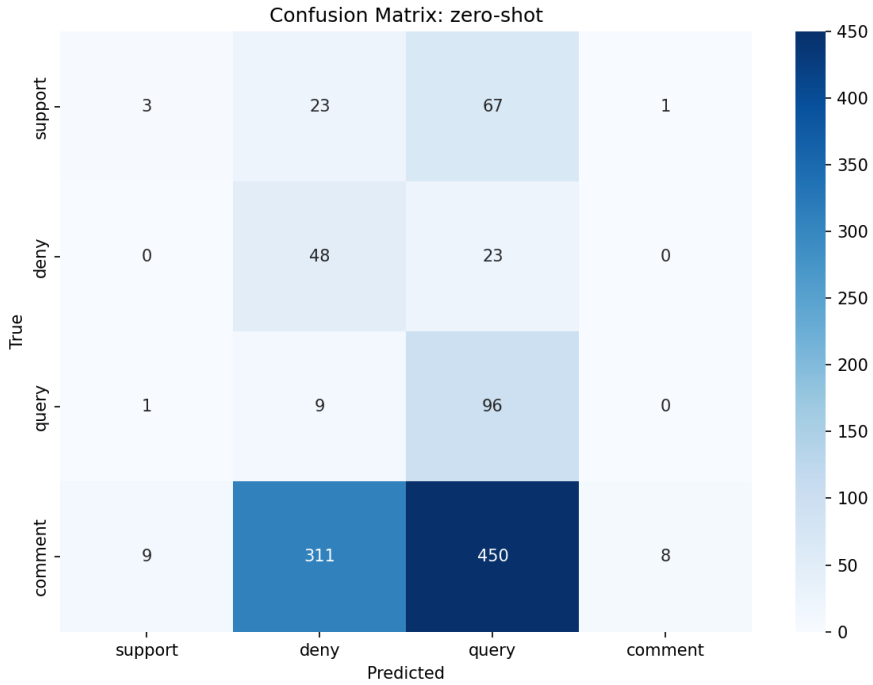

Figure 10: Confusion matrix on the test set for zero-shot basic prompting.

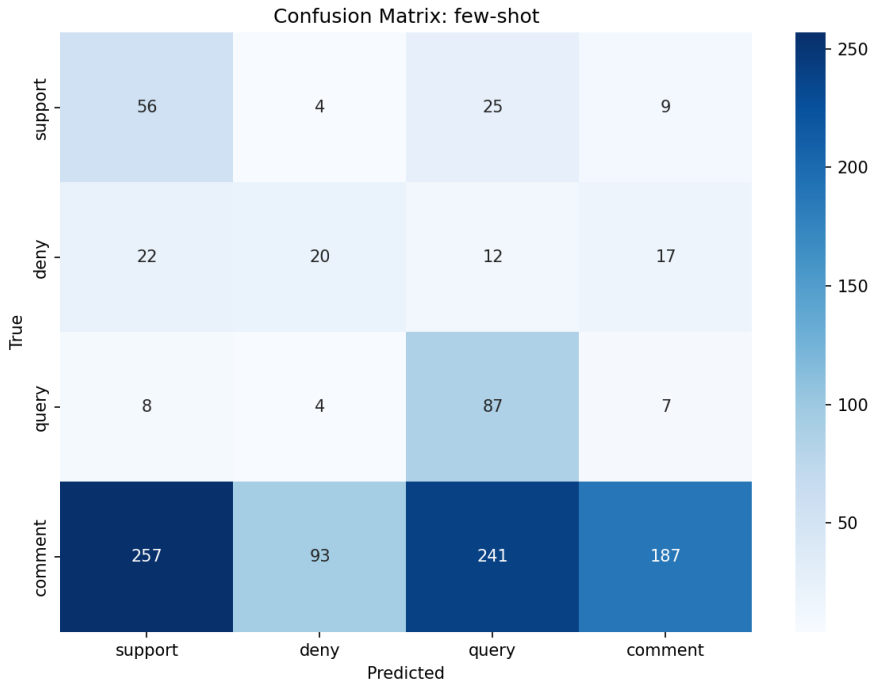

Figure 11: Confusion matrix on the test set for few-shot basic prompting.

Table 6 shows that the macro-F1 was 0.14 for zero-shot and 0.30 for few-shot prompting, so the reasoning examples significantly improve performance as expected. However, there is clearly a severe bias problem with the setup. In zero-shot, the model almost never predicts comment (1% recall) despite it being over 70% of the data! Instead, it massively over-predicts query (91% recall) and deny (68% recall), resulting in very low macro-F1. This is somewhat improved by few-shot examples, as the model begins to predict support and comment more often. However, query is still predicted far too often (82% recall). This suggests the model is defaulting to query. In future, we will test alternative system prompts (e.g. improved label definitions) to investigate if this reduces query over-prediction. Moreover, we test how the stances included in the few-shot examples affects performance, and we find that 1-shot with a query example may be as good as 4-shot with an example from each stance (Figure 12). This surprising result suggests the model is especially prone to predict query.

3.3 CoT Prompting

Chain-of-Thought (CoT) prompting lets models use intermediate reasoning steps before answering, often improving performance [18]. We test zero- and few-shot CoT prompting by asking models to first classify into stance vs non-stance, then classify the stance. Figure 13 shows our exact prompts.

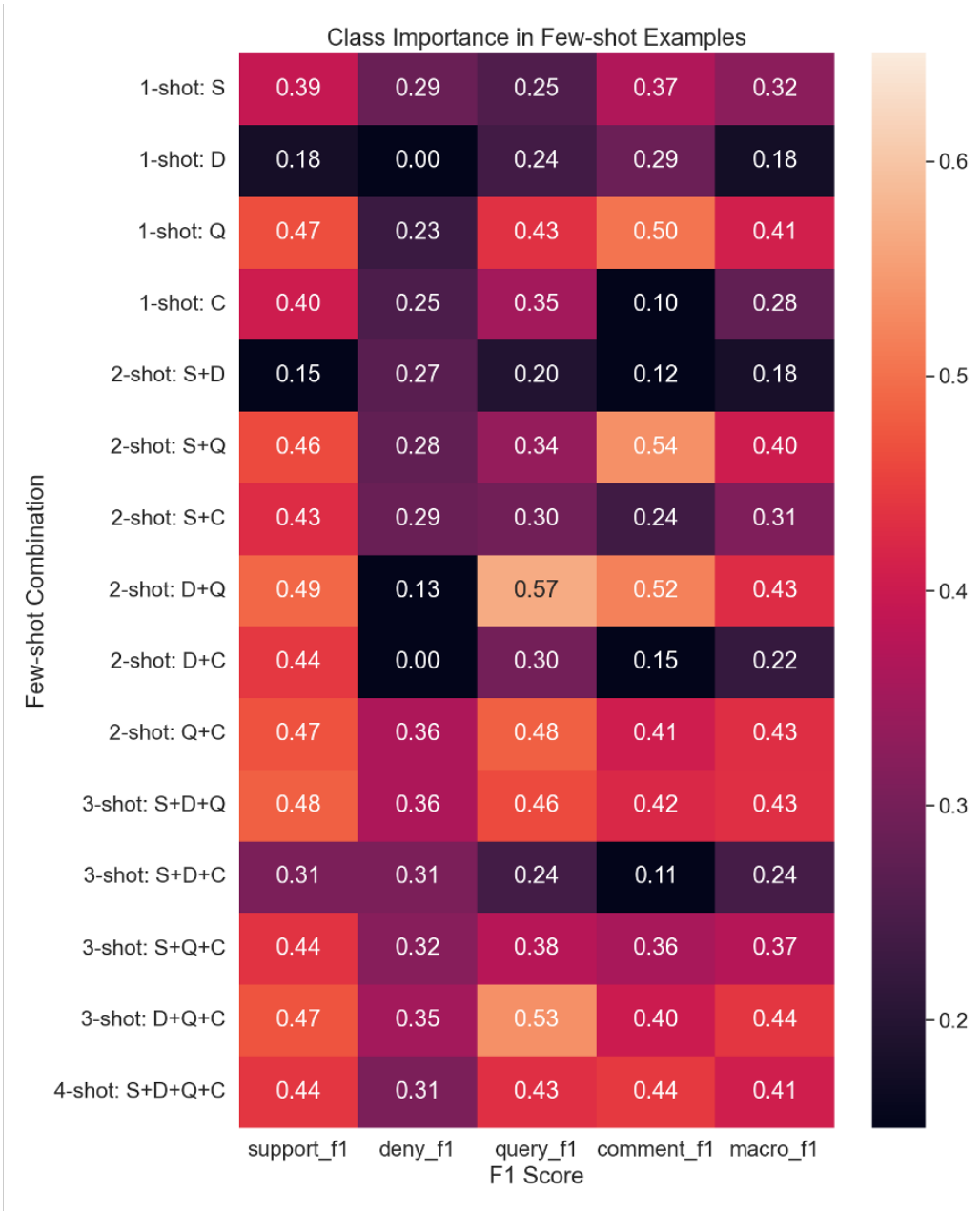

Figure 12: Heatmap showing the impact of different few-shot example combinations on macro and per-class validation F1 scores. These results are from 1 run, and are intended as a starting point for future investigation.

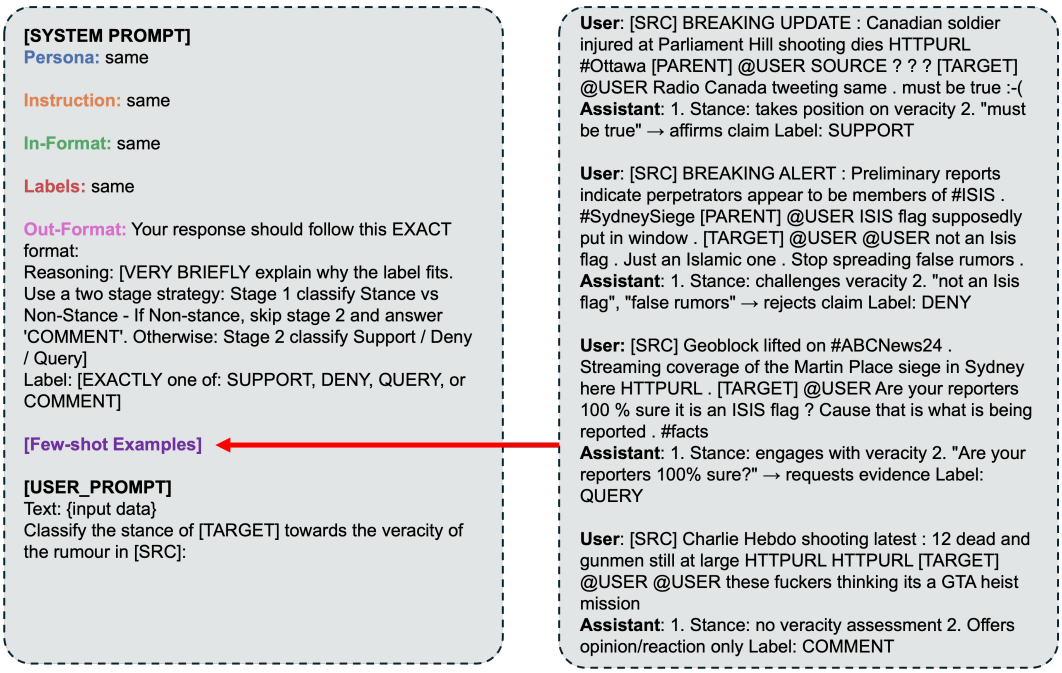

Figure 13: Exact CoT prompt template with the few-shot examples used in our experiments. System prompt is the same as basic prompting, besides the new out-format component.

Table 7: Side-by-side comparison of classification performance on the test set using CoT prompting. P = precision, R = recall.

(a) Zero-shot CoT report.

Stance	P	R	F1	Support
support	0.14	0.29	0.19	94
deny	0.16	0.38	0.22	71
query	0.36	0.45	0.40	106
comment	0.78	0.55	0.65	778
accuracy			0.51	1049
macro avg	0.36	0.42	0.37	1049
weighted avg	0.64	0.51	0.55	1049

(b) Few-shot CoT report.

Stance	P	R	F1	Support
support	0.14	0.55	0.23	94
deny	0.18	0.59	0.28	71
query	0.39	0.68	0.49	106
comment	0.84	0.28	0.42	778
accuracy			0.37	1049
macro avg	0.39	0.53	0.35	1049
weighted avg	0.69	0.37	0.40	1049

Figure 14 shows that the fine-tuned BERTweet classifier outperforms all prompting methods. This was expected, because: 1) the prompting model classified tweets as stance too easily (low comment recall), and 2) fine-tuning lets the model learn to improve, which prompting does not. Moreover, fine-tuning takes ∼30min to run, which is slower than basic prompting (∼15min each) but much faster than CoT prompting (∼1hr each). This suggests our fine-tuning setup is much better for classification than our prompting setup. However, prompting performance heavily depends on prompt design, so future work will test other prompts for enhanced performance.

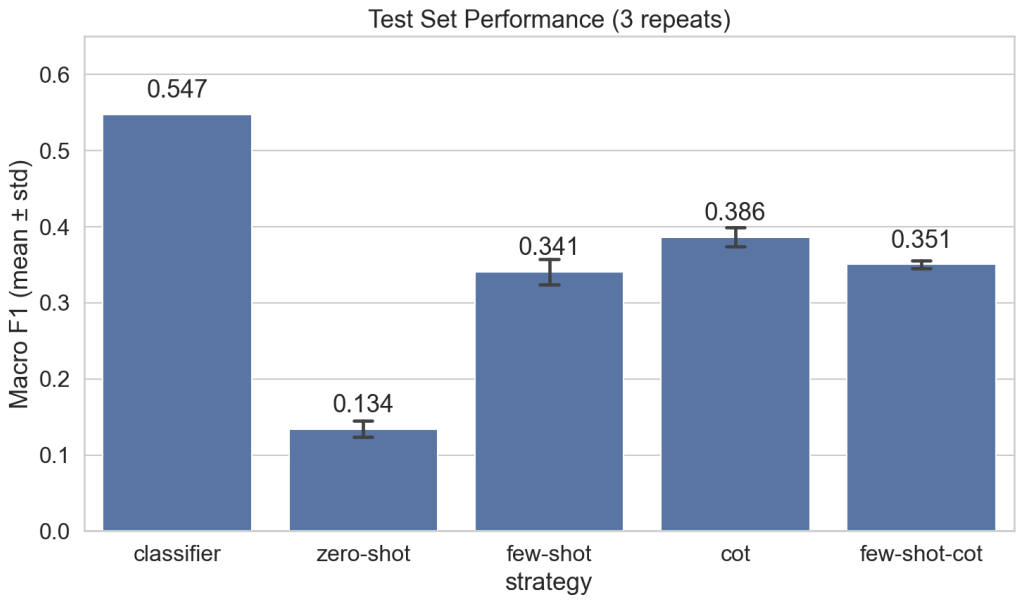

Figure 14: Comparison of all 5 classification methods on the test set. The classifier score is the test set performance of the best model for validation macro-F1 (as discussed in Section 3.1). Prompting scores are averaged over 3 runs on the test set.

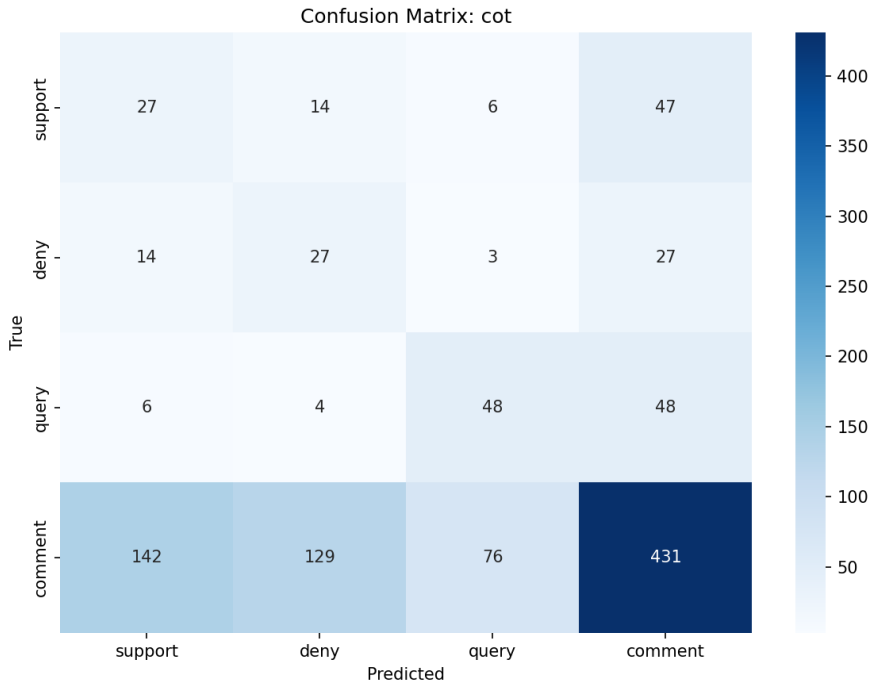

Figure 15: Confusion matrix on the test set for zero-shot CoT prompting.

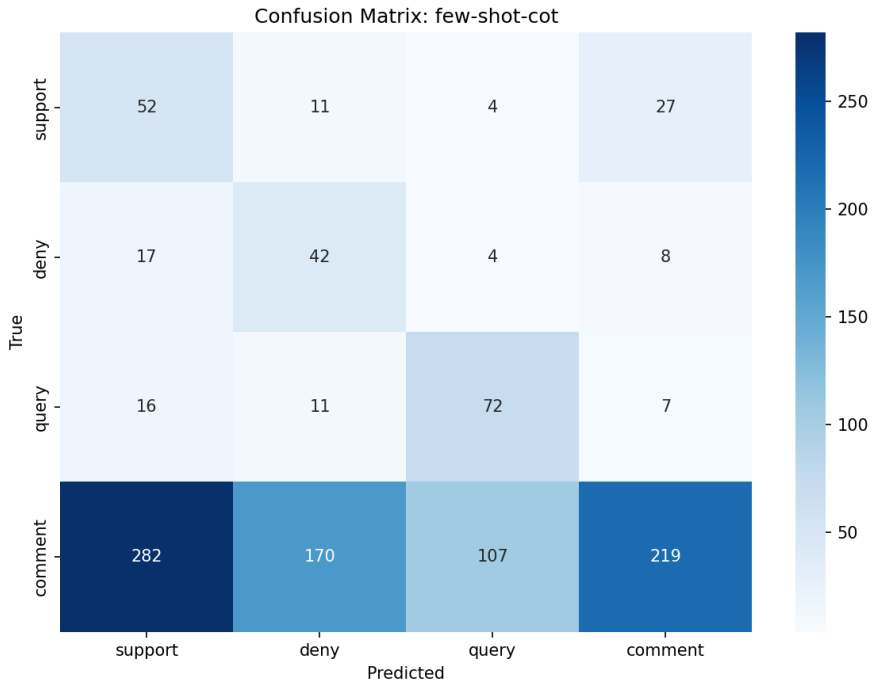

Figure 16: Confusion matrix on the test set for few-shot CoT prompting.

Zero-shot-CoT is best among the prompting methods, suggesting that the few-shot-CoT examples actually misled the model. Table 7 shows that, compared to basic prompting, CoT improves the precision for deny (0.18 few-shot-CoT vs 0.17 few-shot) and query (0.39 few-shot-CoT vs 0.24 few-shot), but performs worse on support (0.14 CoT vs 0.23 zero-shot), so our CoT prompts have not significantly improved minority class precision. Error propagation was a serious issue for CoT prompting. The model tended to classify text as SDQ if it had any sentiment at all, even if it was a comment: e.g. *"So sad! I hope they're okay!"*. This was evidenced by CoTs like: *"1- the author is sad, so this is a stance tweet. 2- ... "*. Such cases were common and were typically classified as support (based on *'hope'*) or query (based on asking if someone is okay). The model clearly misunderstood the task of classifying stance towards rumour veracity, rather than just sentiment detection. Future work will try alternative system prompts and few-shot examples to address this misunderstanding.

4 Ethical Implications

The dataset contains thousands of personal accounts, which presents privacy risks: GDPR codifies the 'right to be forgotten', which would require erasing someone's tweet from twitter and the RumourEval dataset (which the user is likely unaware of). Moreover, it is often impractical to erase it from a model's weights that was fine-tuned using it. We can re-tune from scratch

which takes under an hour, but more intensive fine-tuning would be much harder. The dataset also covers highly sensitive topics, so it must be used with appropriate empathy and consideration of stakeholders.

Our solutions sometimes show high false positive rates. In real-world situations, this could cause users to be falsely flagged as supporting false rumours and unfairly suspended. Automated rumour verification with these faulty systems could amplify false rumours or suppress sincere discussion, and they could be used to adversarially create false rumours that avoid detection. Therefore, these systems require effective human oversight; they should supplement and not replace existing rumour verification methods. Explainability and interpretability tools must be used both during deployment (e.g. CoT monitoring [9]) and beforehand (e.g. sparse autoencoders to interpret model internals [2]).

5 Conclusion

In this paper, we compared a fine-tuned BERTweet and various Llama-3.2-3b-Instruct prompting strategies for stance classification. We found BERTweet performed best, but we will explore CoT prompting in future work since it provides interpretability benefits such as CoT monitoring.

References

[1] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–236, 2017.

[2] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[3] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, editors, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2006. URL https://aclanthology.org/S17-2006/.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[5] Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. Semeval-2019 task 7: Rumoureval 2019: Determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation: NAACL HLT 2019*, pages 845–854. Association for Computational Linguistics, 2019.

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

[8] Elena Kochkina, Maria Liakata, and Isabelle Augenstein. Turing at SemEval-2017 task 8: Sequential approach to rumour stance classification with branch-LSTM. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, editors, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 475–480, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2083. URL `https://aclanthology.org/S17-2083/`.

[9] Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, et al. Chain of thought monitorability: A new and fragile opportunity for ai safety. *arXiv preprint arXiv:2507.11473*, 2025.

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[12] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-

the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

[13] 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.

[14] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2): 945–959, 06 2000. ISSN 1943-2631. doi: 10.1093/genetics/155.2.945. URL https://doi.org/10.1093/genetics/155.2.945.

[15] Rob Procter, Farida Vis, and Alex Voss. Reading the riots on twitter: methodological innovation for the analysis of big data. *International journal of social research methodology*, 16(3):197–214, 2013.

[16] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

[17] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[18] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[19] Brandon T Willard and Rémi Louf. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*, 2023.

[20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

[21] Ruoyao Yang, Wanying Xie, Chunhua Liu, and Dong Yu. Blcu_nlp at semeval-2019 task 7: An inference chain-based gpt model for rumour evaluation. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 1090–1096, 2019.

[22] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3): e0150989, 2016.