NapierNLP at Touché: Fallacy Detection with Small Language Models and Model Fusion

Touché at CLEF 2026

Katarina Alexander[1,*], Md Zia Ullah[1] and Dimitra Gkatzia[1]

[1]School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Scotland

Abstract

This paper presents the participation of our team, NapierNLP, in the CLEF2026 Touché Lab: Fallacy Detection, sub-task 1. This task aimed to identify fallacious statements within a corpus sourced from Reddit comments. We formulate the task as a classification problem and fine-tune four pre-trained small language models (SLMs): GPT2, Qwen, TinyLlama, and GPT-Neo. To enhance our predictions, we combine the outputs from multiple models using two different methods: confidence and majority voting. We conducted experiments and evaluated our approach using the Touché 2026 Fallacy Detection dataset. Our best model produced an F1 result of 0.91, which placed 8th in the competition.

Keywords

Fallacy Detection, Natural Language Processing (NLP), Large Language Models (LLMs), Small Language Models (SLMs),

1. Introduction

Fallacies are arguments that seem convincing, however, are based on faulty reasoning. They are frequently employed in online discourse, which can get argumentative or manipulative. Learning the ability to debate has been shown to decrease verbal aggressiveness [1], therefore, identifying these arguments online can help keep discussions civil. It is not only useful for users of online discussion forums, but also for moderators. By identifying those exchanges that are becoming fallacy heavy moderators can step in and divert the discussion before it becomes too heated, or identify cases where misinformation is being spread.

The CLEF-2026 Touché[2][3] Lab "Fallacy Detection" task invites participants to develop a system that can identify a fallacy, which is "a flawed argument, where the inference relation or the premises are incorrect." [4] From the dataset three subtasks were defined: Fallacy Detection, Fallacy Classification (for fallacious arguments), and Argument Scheme Classification (for non-fallacious arguments). We participated in subtask one only. We utilised small (<5B) parameter versions of open-source large language models to detect fallacies, a continuation of our previous work at the CLEF 2025 CheckThat! competition [5].

2. Background and Related Work

Fallacy detection can be traced back to Ancient Greece with Aristotle's "On Sophistical Refutations" [6], and has since been an important philosophical topic. The dataset used for this competition has it's roots in the Pragma-dialectical theory by van Eemeren and Grootendorst [7], and subsequent work by Walton [8]. Although these definitions are an integral piece of fallacy detection, dataset creation can be a difficult task. Sahai et al [4] followed on from work by Habernal et al. [9][10] and Da San Martino [11]

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

katarina.alexander@napier.ac.uk (K. Alexander); m.ullah@napier.ac.uk (M. Z. Ullah); d.gkatzia@napier.ac.uk (D. Gkatzia)

https://www.napier.ac.uk/people/katarina-alexander (K. Alexander); https://www.napier.ac.uk/people/md-zia-ullah (M. Z. Ullah); https://www.napier.ac.uk/people/dimitra-gkatzia (D. Gkatzia)

0000-0002-4022-7344 (M. Z. Ullah); 0000-0001-8568-7806 (D. Gkatzia)

to create a large dataset of both fallacious and non-fallacious statements. Using a Multi-Granularity Network (MGN) an F1 of 0.7586 was achieved on the full dataset. Large Language Models (LLMs) can be effectively used for language-based classification tasks [12], including fallacy detection. Pan et al [13] achieved an F1 of 0.8111 on the same dataset using GPT-4.

Using an ensemble model can yield better results than individual models [14]. Hard voting is a straightforward technique for classification problems, where individual models classify sentences independently, and the final prediction is based on the most frequently assigned label [15]. Whilst hard voting is done post classification, soft voting is done prior to label creation. Predicted probabilities from the different models are combined and then a label is assigned based on the average, giving models with a higher confidence a greater impact on the final label [16].

3. Methodology

For this experiment we framed fallacy detection as a binary classification problem. Models were fine-tuned using supervised learning of human-annotated data to classify sentences accordingly.

3.1. Small Language Models

For the purposes of this experiment, we define Small Language Models (SLMs) as pre-trained neural language models with under 5B parameters. SLMs are a subsection of Large Language Models which have emergent abilities that are not present in other architectures [17], including the ability to be prompted with natural language. As well as prompting, SLMs can also be fine-tuned for a variety of downstream tasks, including sentence classification.

In this study, we fine-tune four publicly available SLMs from HuggingFace: GPT2 [18], Qwen2.5 [19, 20], TinyLlama [21] and GPT-Neo [22] and two encoder-only transformers with bi-directional attention: Bert[23] and Roberta[24]. Details of the models can be found in Table 1.

Table 1
Details of the models considered in the experiments and evaluation

Model	Parameters	Developer	Release Year	Available at:
GPT2	137M	OpenAI	2019	openai-community/gpt2
GPT-Neo 1.3B	1.37B	EleutherAI	2021	EleutherAI/gpt-neo-1.3B
Qwen2.5 1.5B	1.54B	Alibaba Group	2024	Qwen/Qwen2.5-1.5B
TinyLlama 1.1B	1.1B	StatNLP Research Group	2024	TinyLlama/TinyLlama_v1.1
BERT		Google	2018	Google/BERT
RoBERTa		FacebookAI (META)	2019	FacebookAI/RoBERTa

We anticipate that a fusion model could outperform individual models, so we employed two different ensemble techniques. In hard (majority) voting each model provides a prediction label, and the common label across all four models is used as the final classification. In soft voting each model produces class confidence probabilities, and the final classification is determined by averaging these probabilities across all models. Therefore we have four different ensemble methods we will try:

HV Hard voting with the four SLMs.

HVB Hard voting with the four SLMs, BERT and RoBERTa.

SV Soft voting with the four SLMs.

SVB Soft voting with the four SLMs, BERT and RoBERTa.

4. Experiments and Evaluations

4.1. Dataset

The corpus consisted of sentences sourced from Reddit [4], an online platform which facilitates anonymous discussion on a plethora of subjects. Each entry contained the following information:

id	Unique identifier for the entry.
text_raw	The original Reddit comment.
text_raw_parent	The parent comment of *text_raw*.
text_raw_title	Title of the thread *text_raw* originates from.
text_base	Self-contained rewrite of *text_raw* that incorporates relevant information from *text_raw_parent* and *text_raw_title*.
argument_base	*{claim, supports}* extracted from text_base, making the argument structure explicit.
text_enhanced	Rewrite of *text_base* that explicitly surfaces fallacy and scheme information.
argument_enhanced	*{claim, supports}* extracted from *text_enhanced*.
fallacy_exists	Binary label (0/1) indicating whether the entry contains a fallacy — **used in Sub-Task 1 (Fallacy Detection)**.
fallacy_type	The specific type of fallacy — **used in Sub-Task 2 (Fallacy Classification)**.
resembles_fallacy	The fallacy type this argument most closely resembles. For fallacious entries, this equals *fallacy_type*. For non-fallacious entries, it identifies which fallacy type the argument could plausibly be confused with — i.e., the argument exhibits a similar reasoning pattern but does not commit the fallacy.
classification	*{argument_goal, argument_basis}* — **used in Sub-Task 3 (Argument Scheme Classification)**.

Fallacious arguments were initially identified through the scraping of any comment that mentioned named fallacies, and then the dataset was cleaned using crowdsourcing on Amazon Mechanical Turk [4]. Non-fallacious comments came from both comments labelled non-fallacious at crowdsourcing, and random picked comments from specific users on Reddit. The complete dataset was evenly distributed between fallacious and non-fallacious arguments, as can be seen in table 2.

Table 2
Statistics of the Train and Test datasets

Dataset	Fallacy	Non-Fallacy	Total
Train	465	473	938
Test	115	118	233

4.2. Experimental settings

We fine-tune the SLMs listed in Table 1, BERT and RoBERTa on the training data for 50 epochs with early stopping using AutoModelForSequenceClassification. Entrants were able to choose between using the base data only, or using the enhanced data. Therefore we ended up with 12 separate trained models, with the base versions only trained on "text_base", "argument_base" and "fallacy_exists" as outlined in section 4.1, and the enhanced versions trained on on "text_enhanced", "argument_enhanced" and "fallacy_exists". We used a parameter-efficient fine-tuning method called LoRA [25] on all models for consistency. The training was conducted on an NVIDIA GeForce RTX 2080 Super with 8 GB GPU memory. TIRA [26] was used for the submission process, with a maximum of three entries per subtask.

4.3. Results and Discussions

Our best result had an F1 of 0.910, which placed 7th in the competition, as seen in table 3. This was the SVB ensemble trained and tested on the enhanced data. Our HVB trained and tested on the enhanced data achieved an F1 of 0.888, which placed 10th in the competition. As our third submission we chose to submit our SVB ensemble trained and tested on the base data only. This placed 7th in comparison to all other submissions also trained and tested on the base data only, with an F1 of 0.734.

Table 3
F1, Precision and Recall results of the top three teams in comparison to our three submitted approaches and the baseline

#	Team	F1	P	R
1	grtsh	0.807	0.807	0.807
2	uzhcl-for-the-win	0.751	0.770	0.754
3	veritas	0.737	0.742	0.737
7	NapierNLP SVB	0.734	0.736	0.734
	Organizer Baseline	0.460	0.502	0.501

(a) Our Soft Voting Ensemble + Berts Trained and Tested on Base Data only.

#	Team	F1	P	R
1	seals-nlp	0.957	0.957	0.957
2	uzhcl-for-the-win	0.944	0.946	0.945
3	ooooooo	0.940	0.940	0.940
7	NapierNLP SVB	0.910	0.910	0.910
10	NapierNLP HVB	0.888	0.888	0.889
	Organizer Baseline	0.438	0.597	0.533

(b) Our Soft Voting Ensemble + Berts and Hard Voting Ensemble + Berts Trained and Tested on Enhanced Data only

Trained and Tested on Base Data: The results of all four SLMs, BERT and RoBERTa and our ensembles trained and tested on the base data only can be found in table 4. Our best individual model was Qwen2.5, with an F1 of 0.751. This was our best overall model, outperforming all our ensemble methods. However, our ensemble methods performed better than the other individual models and adding BERT and RoBERTa improved the performance of both our soft and hard voting ensembles for the Binary F1 score. However, for the Macro F1 score the HV ensemble (without either BERT or RoBERTa) was the best ensemble, and would have placed 3rd in the competition. Further work on which models to use for the ensemble is needed: a quick experiment just using Qwen and TinyLlama increased the Binary F1 of the SV model to 0.754 and the HV model to 0.751.

Table 4
Performance of models on the test dataset: Binary F1, Precision, Recall, Accuracy, and Macro F1 - Trained and tested on the base data only.

Model	Binary F1	Precision	Recall	Accuracy	Macro F1
GPT2	0.686	0.654	0.722	0.674	0.673
Qwen2.5	**0.751**	0.730	**0.774**	**0.747**	**0.747**
GPT-Neo	0.647	**0.742**	0.574	0.691	0.686
TinyLlama	0.713	0.674	0.757	0.700	0.699
Soft Voting	0.737	0.719	0.757	0.734	0.734
Hard Voting	0.744	0.731	0.757	0.743	*0.743*
Bert	0.645	0.602	0.696	0.622	0.621
Roberta	0.698	0.636	**0.774**	0.670	0.667
Soft Voting + Berts	0.742	0.712	**0.774**	0.734	0.734
Hard Voting + Berts	*0.745*	0.718	**0.774**	0.738	0.738

Trained and Tested on Enhanced Data: Our models trained and tested on the enhanced data can be seen in table 5. The enhanced data improved the results by an average of 0.17 for the F1 score. The best performing model was TinyLlama, which was also our best performer overall with an F1 of 0.914. In

contrast to the base dataset, a soft voting ensemble worked better than a hard voting ensemble, and adding BERT and RoBERTa to the hard voting ensemble reduced its efficiency.

Table 5
Performance of models on the test dataset: Binary F1, Precision, Recall, Accuracy, and Macro F1 - Trained and tested on the enhanced data only.

Model	Binary F1	Precision	Recall	Accuracy	Macro F1
GPT2	0.850	0.816	0.887	0.846	0.845
Qwen2.5	0.897	**0.926**	0.870	0.901	0.901
GPT-Neo	0.827	0.769	0.896	0.816	0.815
TinyLlama	**0.914**	0.906	**0.922**	**0.914**	**0.914**
Soft Voting	0.900	0.897	0.904	0.901	0.901
Hard Voting	0.900	0.904	0.896	0.901	0.901
Bert	0.835	0.835	0.835	0.837	0.837
Roberta	0.872	0.884	0.861	0.876	0.876
Soft Voting + Berts	*0.910*	0.898	**0.922**	*0.910*	0.910
Hard Voting + Berts	0.888	0.880	0.896	0.888	0.888

4.4. Failure Analysis

Although our model had a high success rate, 14 of the 233 sentences were misclassified by all eight ensemble methods, as seen in table 6. Of the different fallacy types, none of the sentences tagged as either being (if a fallacy) or resembling (if not a fallacy) black-or-white thinking were misclassified. The most common fallacy types to be misclassified as non fallacies were as an appeal to authority (somebody important said it was the case) and appeal to worse problems (problem X is worse, so the original problem is justified). The most common non-fallacy to be misclassified as a fallacy resembled the appeal to tradition fallacy (we have always done this, so this is correct). In the case of entry 1324: 'Have we just won the right to use the argument "we've been doing VEGANISM since literal ages" now?' it is likely because it does contain the fallacy, but it is not being used as one.

Table 6
Sentences which all ensemble methods failed to classify correctly: their gold standard label and which fallacy they represent or are similar to

#	First ten words of text	Gold Label	Fallacy Type
44	Because it's a player shitting on the media, and this	1	authority
88	Using your IQ as an argument to validate your point.	1	authority
201	Now let us look at Ghostbusters and how did that	1	hasty generalization
454	If you allow complex bans, you can 'complex ban' almost	1	slippery slope
634	Reddit Moderator Shitshow Recap: They messaged us on the behalf	1	worse problems
641	I really don't like this response, this is giving some	1	worse problems
761	Nationwide regulation has nothing to do with whether or not	0	authority
1070	Everyone's ancestor has taken someones land at this point. If	0	natural
1115	Based on your username, I'm going to say this is	0	population
1257	This is a very slippery slope. What's next to be	0	slippery slope
1293	I think I might be the only person not looking	0	tradition
1296	That filth pays for the services that this province provides	0	tradition
1324	Have we just won the right to use the argument	0	tradition
1422	Fuck! I will never complain about my life ever again.	0	worse problems

Of the correctly identified sentences, 155 of 233 were correctly identified by every single ensemble method, and 205 were correctly identified by every single enhanced data ensemble. When split by fallacy type, the enhanced data ensembles also correctly identified every appeal to nature fallacy and every

non-fallacy resembling hasty generalization on top of the black-or-white thinking fallacies mentioned previously. Only two sentences were correctly identified by all base ensembles and misidentified by all enhanced ensembles: 401 (fallacy) and 1403 (non-fallacy). This may be due to the fact the text is longer in the enhanced_text field than in the base_text field, however, as SLMs are black-box algorithms further investigation and experimentation would be needed to identify the cause.

5. Conclusion

For our approach, we tested four different small language models, BERT and RoBERTa and used two different ensemble methods to classify the sentences as either a fallacy or a non-fallacy. For the models trained and tested on the base data only we found that Qwen2.5 was both the best individual and best overall model. We also found that the hard voting ensemble performed better than the soft-voting ensemble, and adding BERT and RoBERTa to the ensemble increased the performance of both ensemble types. In contrast, for models trained and tested on the enhanced data, TinyLlama was the best individual and overall model. The soft voting ensemble outperformed the hard voting ensemble both with and without BERT and RoBERTa, whose addition actually reduced the hard voting ensemble F1 score.

6. Future Work

There are several ways in which this research could be enhanced in future work. For example, identifying other small language models or BERT-based models which could be included in the ensembles, or removing models that may reduce the overall ensemble performance. Further investigation of ensemble methods may also result in a higher score, for example, by including a weighting to ensure models such as Qwen2.5 or TinyLlama have a larger impact on the overall ensemble. Re-training the models to access more of the data fields available, such as the raw text, parent and title may also have improved both the individual models and therefore the ensembles F1 scores. Additionally, we could investigate prompting, either as an alternative to or in conjunction with our current method.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] J. A. Sanders, R. L. Wiseman, R. H. Gass, Does teaching argumentation facilitate critical thinking?, Communication Reports 7 (1994) 27–35. URL: https://doi.org/10.1080/08934219409367580. doi:10.1080/08934219409367580. arXiv:https://doi.org/10.1080/08934219409367580.

[2] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, C. Mathews, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, Overview of touché 2026 — argumentation systems, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. S. Salido, A. Barrón-Cedeño, A. G. S. Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. 17th International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[3] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, C. Mathews, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, Overview of touché 2026 — argumentation systems — extended version, in: E. S. Salido, A. Barrón-Cedeño, A. G. S. Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[4] S. Sahai, O. Balalau, R. Horincar, Breaking down the invisible wall of informal fallacies in online discussions, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 644–657. URL: https://aclanthology.org/2021.acl-long.53/. doi:10.18653/v1/2021.acl-long.53.

[5] F. Alam, J. M. Struß, T. Chakraborty, S. Dietze, S. Hafid, K. Korre, A. Muti, P. Nakov, F. Ruggeri, S. Schellhammer, et al., Overview of the clef-2025 checkthat! lab: Subjectivity, fact-checking, claim normalization, and retrieval, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2025, pp. 199–223.

[6] W. A. Pickard-Cambridge, Topics, in: J. Barnes (Ed.), Complete Works of Aristotle, Volume 1: The Revised Oxford Translation, Princeton University Press, 1985, pp. 167–277.

[7] F. H. van Eemeren, R. Grootendorst, Speech Acts in Argumentative Discussions: A Theoretical Model for the Analysis of Discussions Directed Towards Solving Conflicts of Opinion, Foris Publications, Dordrecht, Netherlands, 1984.

[8] D. Walton, G. Restall, Justification of argumentation schemes, Argumentation 27 (2005).

[9] I. Habernal, R. Hannemann, C. Pollak, C. Klamm, P. Pauli, I. Gurevych, Argotario: Computational argumentation meets serious games, in: L. Specia, M. Post, M. Paul (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 7–12. URL: https://aclanthology.org/D17-2002/. doi:10.18653/v1/D17-2002.

[10] I. Habernal, H. Wachsmuth, I. Gurevych, B. Stein, Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 386–396. URL: https://aclanthology.org/N18-1036/. doi:10.18653/v1/N18-1036.

[11] G. Da San Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, P. Nakov, Fine-grained analysis of propaganda in news articles, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5636–5646. URL: https://aclanthology.org/D19-1565/. doi:10.18653/v1/D19-1565.

[12] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, A. Mian, A comprehensive overview of large language models, 2024. URL: https://arxiv.org/abs/2307.06435. arXiv:2307.06435.

[13] F. Pan, X. Wu, Z. Li, A. T. Luu, Are LLMs good zero-shot fallacy classifiers?, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 14338–14364. URL: https://aclanthology.org/2024.emnlp-main.794/. doi:10.18653/v1/2024.emnlp-main.794.

[14] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, 1st ed., Chapman & Hall/CRC, 2012.

[15] Fusion of Label Outputs, John Wiley & Sons, Ltd, 2004, pp. 111–149. doi:https://doi.org/10.1002/0471660264.ch4.

[16] C. Ju, A. Bibaut, M. J. van der Laan, The relative performance of ensemble methods with deep convolutional neural networks for image classification, arXiv preprint arXiv:1704.01664 (2017).

[17] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, W. Fedus, Emergent abilities of large language models, 2022. arXiv:2206.07682.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners (2019).

[19] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei,

H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Fan, Qwen2 technical report, arXiv preprint arXiv:2407.10671 (2024).

[20] Q. Team, Qwen2.5: A party of foundation models, 2024. URL: https://qwenlm.github.io/blog/qwen2.5/.

[21] P. Zhang, G. Zeng, T. Wang, W. Lu, Tinyllama: An open-source small language model, 2024. URL: https://arxiv.org/abs/2401.02385. arXiv:2401.02385.

[22] S. Black, G. Leo, P. Wang, C. Leahy, S. Biderman, GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, 2021. URL: https://doi.org/10.5281/zenodo.5297715. doi:10.5281/zenodo.5297715.

[23] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, W. Chen, Lora: Low-rank adaptation of large language models, CoRR abs/2106.09685 (2021). URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

[26] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous integration for reproducible shared tasks with tira.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.