AIvengers at SemEval-2026 Task 9: Utilizing Language Specific Encoders for Multilingual Text Classification

Lars Britz
University of Augsburg, Germany
lars.britz@uni-a.de
Boon Elschenbroich
University of Augsburg, Germany
boon.elschenbroich@uni-a.de

Abstract

Polarizing language has evolved from a social media phenomenon into a pervasive feature of public and everyday discourse across cultures and geographies. And, this is not limited to certain countries, but a world wide trend. As we will show, detecting polarization, it's type and manifestation is not a simple task for one ML model, but, it requires multiple different approaches depending on the language and culture. In this paper, we provide the best methods that we found for each language in all three SemEval 2026 - Task 9 multilingual text classification challenge subtasks. We achieved the best results with language specific pre-trained BERT and RoBERTa models, rather than using a general approach and using a generic multi-language model. Our approach secured a high to medium rank in all subtasks and languages.

1 Introduction

The SemEval-2026 Task 9 (Naseem et al., 2026a) consists of three classification sub-tasks over 22 languages: (subtask 1) Polarization detection (binary classification), (subtask 2) Polarization type classification and (subtask 3) Manifestation identification (both multi-label classification). Multiple labels could be assigned per statement for subtask 2 and 3. Training data was provided for all languages in Subtasks 1 and 2, but Subtask 3 did not provide data for five languages.

The 22 languages covered by Task 9 (Naseem et al., 2026b) were: Amharic (amh), Arabic (arb), Bengali (ben), Burmese (mya), English (eng), German (deu), Hausa (hau), Hindi (hin), Italian (ita), Khmer (khm), Nepali (nep), Odia (ori), Persian (fas), Polish (pol), Punjabi (pan), Russian (rus), Spanish (spa), Swahili (swa), Telugu (tel), Turkish (tur), Urdu (urd), and Chinese (zho). A fully labeled training set and an unlabeled development set (verifiable by uploading results to Codabench [1])) were available for the initial development phase.

[1] https://codabench.org

2 Related work

Detecting polarization and hate speech in statements was never more important than today. The classification of polarization is very closely related to sentiment analysis, and there is quite a lot of research available for sentiment analysis. There are many NLP approaches available for different languages, either BERT-based (Camacho-collados et al., 2022; ssary, 2021), or LLM-based (Zhang et al., 2024; Wang et al., 2024; Hofmann and Friedrich, 2025; Sun et al., 2023).

The challenge is that there are 22 languages in the tasks, and some of them are not very widely used. There are not many pre-trained models available for such languages, and LLM models also struggle with them. (Dementieva et al., 2025) is providing an approach to handle classification in such languages.

Other publications also provide more details on different scripts and how they can influence classification results (Abdullah et al., 2025). This is especially the case for languages based on Arabic script.

With the task at hand, we will also need some good approach to finetuning models based on small datasets. The challenge of this is explained in (Mosbach et al., 2021) and (Zhang et al., 2021).

3 Data

With statements coming from 22 different languages and cultures, labeling had to be done from different groups of people coming from different cultures and having different native tongues. The first analysis that we performed was therefore to investigate how heterogeneous the labeling would be in the different languages.

Even in subtask 1, we could see some misalignment between the languages (cf. fig. 1). While the

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 761–776

polarization rate in most languages was around 50 percent, some languages showed extremes, such as very low polarization in Hausa and very high polarization in Khmer. This might be due to cultural or political characteristics in these languages and countries, or it might just show a lack of quality in the labels.

In subtask 2 and 3 (cf. fig. 2 and 3) we could observe similar outliers. Khmer again showed just very few labels set over all, whereas other languages, such as Urdu show many labels set, even often multiple ones on the same statements.

As the results later demonstrate, the sparse label distribution in Hausa renders reliable multi-label classification particularly challenging.

4 Methods and experiments

4.1 Overview

The first subtask is a simple binary classification; the other two subtasks are multi label classifications with labels occurring more than once per item. The classification strategy must be aligned accordingly.

We followed an approach to empirically determine the best approach and model for the language and subtask.

4.2 Data pre-processing

The records in the training, dev, and test data set were taken as is from the social media occurrence to the provided files, including line breaks and emojis. In order to remove the effect of emojis on the training and prediction, we implemented a method and option to filter emojis.

4.3 Subtask 1

Subtask 1 is a simple binary classification. The goal is to detect polarizing speech in given statements. In order to find a good solution, we evaluated different approaches.

We evaluated BERT and RoBERTa variants as embedding backbones, with a task-specific neural network head for classification. This was done using pre-trained models. We also tried different training options, with different hyper training parameters and different strategies.

We also used LLM models for the polarization detection for some languages, but, this approach is not generally recommended, because low-resource languages are not handled well in available LLMs. The approach also takes more time and resources and is therefore less efficient.

4.3.1 BERT plus NN

For the BERT models, we used the libraries auto tokenizer and auto model libraries provided by HuggingFace's transformer libraries (Wolf et al., 2020).

The BERT model's embedding output was then fed into a three layer neural network that then leads to the classification for the 2 classes: polarizing and non-polarizing.

Because for some of the languages the labels were not balanced, we also added an option for a weighted training, feeding the weights into a cross entropy loss function.

Each training was set up so that an evaluation was done after each epoch. The best result was saved to disk and later loaded for the final predictions.

First, we tested with generic multi-language models only, such as FacebookAI's xlm-roberta-large (Conneau et al., 2019). It quickly became clear that models, which were pre-trained on the specific language, performed considerably better.

To find the right models, we used the search on the HuggingFace webpage. [2]. We searched for the language or region for BERT based models. A broad search provided a list of models that we tested and later used (cf. appendix A.4).

For some languages, especially in the subtask 1, we also used models that were pre-trained for sentiment analysis, which is very similar to the polarization task. These models usually performed better than models that were more focused on general purpose.

Single run - multiple epochs

What we tested and ended up using for most languages is a single training run with a random 80/20 data split. The entire model was updated during this model training.

For most languages, we used 20 epochs. For some languages, we could see that we reached the best results much earlier in our test runs, so we reduced the number of epochs for those languages. But also sometimes we could observe improvements even on later epochs.

For the majority of languages, this approach, together with a pre-trained model, gave us the best results.

Training with frozen layers

We also tried the option to not train the full model

[2] https://huggingface.co/models

on all epochs, but to split the training cycles into phases, where we could only train the neural network, or finetune the BERT model, or train both.

The results were not much different from the other options that we explored; therefore, we did not use this approach on the final prediction runs.

5-fold cross validation

There are not too many training statements available (approximately 3,000 per language). In a normal split training, such as 80/20, we would lose precious data to train on. This is why we also used 5-fold cross validation training for some languages. Not all predictions improved by doing this. For some languages, we did not see much difference. However, for a few languages, we did see improvements and used this method during training.

Paraphrase generation

Because we had fewer data records than we would like to have, we explored the possibility o using the existing records to create new records by paraphrasing the provided ones. We used a T5 (Raffel et al., 2020) model to perform the extra data augmentation (Kumar et al., 2020) (model name: Vamsi/T5_Paraphrase_Paws (Zhang et al., 2019)).

We tried this approach in English and German. It did not improve the results, the training only took longer because of the additional data.

4.3.2 LLM

We also tried an LLM approach for classification, just asking the LLM to classify the data for us. We used different prompts and measured the results. We ended up using this approach for some languages. But we used it in combination with different approaches (cf. chapter 4.3.3).

Zero shot prompt

We first tried the naïve approach, just asking the LLMS if the single sentence is polarizing. Results were weak, barely exceeding the random baseline.

The approach took much longer than training a BERT + NN model. We expected some better results from this approach.

We ran this experiment with two very common models Qwen2.5-7B-Instruct (Yang et al., 2024) and Llama-3.1-8B-Instruct (Llama Team, AI @ Meta, 2024) that could comfortably run on the "consumer" hardware that we used.

Few shot prompt

With seeing just bad results using a zero shot approach, we switched to a few shot approach. The prompt can be found in the appendix A.6.

We provided 10 random samples for polarizing statements and 10 random samples for non-polarizing samples.

Using this approach immediately provided better results than the zero shot prompt. The predictions were comparable with the results from the BERT + NN approach.

We also used the same two models Qwen2.5-7B-Instruct (Yang et al., 2024) and Llama-3.1-8B-Instruct (Llama Team, AI @ Meta, 2024) for these predictions.

4.3.3 Hybrid approach with majority vote

While we could see good results for many languages with a simple BERT + NN model and a proper pre-trained BERT or RoBERTa model, we saw results that were not comparable to the best results that we saw on Codabench. This was especially true for the languages German and English, which is surprising, because most of the models should perform very well for these two high-resource languages.

In order to improve the results, we thought about a different strategy, where we used different models and approaches to come up with predictions, and then took a majority vote on the results for a combined predictions. We used odd numbers of predictions to get a more robust result.

We not only used BERT models for this, but also combined the BERT based predictions with the output from LLM predictions.

With this approach, we were able to raise the F1 macro score for the English dev set from 0.7859 to 0.8276, which, at the time of submission was among the 10 best results.

4.3.4 Adapting learning rate

In addition to all the high-level optimizations, we also tried to optimize on a lower level. We mostly experimented with the learning rate. These changes and experiments were evaluated on the training performance and not on the submitted dev data sets. And because we only saw marginal differences by choosing other learning rates, we ended up using a default learning rate of 2e-5 for all model training.

4.3.5 Strategy and model per language

As already mentioned, the best method and model for each language was determined empirically by

experiments. For some languages, we saw very good results without giving it a second thought. Other languages were more difficult to work with. For several languages, we tried various models and model families, to find the best solution.

The appendix A.8 contains all parameters used for training for each model and language for subtasks 1.

4.4 Subtask 2 and 3

Regarding the multi-label classification tasks, we explored three approaches for the identification of manifestations or types of polarization. The two main approaches leverage a pre-trained transformer-based language model as the backbone, utilizing multilingual models such as XLM-RoBERTa(Conneau et al., 2019) or for more difficult languages, specialized transformers (see A.8). We also tried a few-shot LLM approach with little success.

4.4.1 Combined multi-label classification (BERT plus NN combined)

Our first approach treats the task as a unified multi-label classification problem. The model utilizes the `[CLS]` token representation from a Transformer encoder, which is projected into an n-dimensional output space via two linear layers, where n denotes the number of target classes. To capture inter-label dependencies, the architecture incorporates a label-correlation layer and is optimized using an asymmetric loss function (Ben-Baruch et al., 2020). To address the inherent class imbalance, we applied a post-hoc threshold optimization strategy. This involves determining the optimal decision threshold for each label independently by maximizing the Macro F1-score on the validation set.

Single run

Similarly to subtask 1, we used a single run training, mainly using 20 epochs and an 80/20 split between training and validation.

Also, for subtask 2, we chose the best model from those 20 epochs. This means that we might have used a model from epoch 5, and discard the remaining training cycles.

5-fold cross validation

For some of the languages, we also saw an improvement using a 5-fold cross validation training. Some languages performed worse using this approach, but few showed better results.

4.4.2 BERT plus NN individual

Our second approach decomposes the multi-label classification task into n independent binary classification problems. This formulation offers the advantage of label specific optimization and allows for targeted feature engineering. However, this approach also presents a disadvantage with respect to data sparsity. For certain languages, the dataset did not contain a single positive instance for specific labels, leading to ineffective models for those categories. The general architecture for these binary classifiers consists of the same pre-trained BERT base and NN head described previously in Section 4.3.1.

4.4.3 Few Shot prompt

Building upon the promising performance of few shot LLMs in our binary classification tasks (Section 4.3.2), we also explored the capabilities of small generative models for multi-label classification. We implemented a few-shot learning pipeline using Llama 3.1 (8B) (Llama Team, AI @ Meta, 2024) to assess whether an LLM could perform effectively. We constructed a prompt using a chain-of-thought reasoning strategy (see Appendix A.7), instructing the model to adhere to a four-step analysis process.

Unfortunately, the model struggled with this complex task and the evaluation on the training set only yielded a Macro F1 score between 0.10 and 0.20. Consequently, we discontinued this approach to focus on the other methods.

5 Results

5.1 Development phase

During the development phase, we tried different models, methods, and approaches to empirically find the best approach for each subtask and language. We were able to come close to or even surpass some of the best results (cf. fig. 4) on Codabench.

It was surprising that we had some good results for low-resource languages where we only had few pre-trained models available.

5.2 Test phase

For the test phase, we combined the publicly released dev set with the original training set to get a larger training data set.

Table 1: Our official F1 scores from the leaderboard in % vs. POLAR Baseline scores in % (PBL) across all subtasks and languages. Green = Ours outperforms POLAR Baseline; Yellow = POLAR Baseline outperforms ours. Dashes indicate languages without annotations for that subtask.

	Subtask 1		Subtask 2		Subtask 3	
Lang.	**PBL**	**Ours**	**PBL**	**Ours**	**PBL**	**Ours**
amh	71.51	77.31	37.16	55.82	44.33	55.35
arb	79.57	81.83	48.55	63.73	39.02	58.53
ben	85.28	83.30	28.87	35.10	8.68	24.42
deu	67.14	70.70	40.78	48.70	34.85	43.65
eng	78.02	81.28	33.33	43.96	41.00	47.32
fas	84.24	80.54	46.26	59.61	20.04	44.37
hau	77.53	81.21	20.38	30.44	74.56	17.81
hin	73.79	78.46	79.11	78.07	23.48	74.88
ita	67.73	64.89	37.59	21.36	—	
khm	65.92	75.31	62.68	64.25	60.95	37.74
mya	82.10	84.43	47.72	63.82	—	
nep	87.98	87.23	72.19	76.95	13.14	64.78
ori	77.65	76.83	56.00	59.38	38.41	25.37
pan	78.98	77.87	36.50	50.31	45.61	52.90
pol	72.41	80.72	44.91	57.52	—	
rus	74.57	75.77	59.04	52.48	—	
spa	72.66	74.97	59.35	61.18	50.88	48.25
swa	75.71	79.56	44.17	47.49	22.05	56.52
tel	64.40	88.64	31.45	44.48	67.38	39.74
tur	69.57	75.98	47.08	59.16	76.93	49.37
urd	78.90	73.14	71.27	77.05	53.16	79.97
zho	86.91	89.10	66.97	76.47	-	63.98
Average		79.05		55.79		49.16

The methods for training the models and producing the labels for the predictions submitted were based on what we had developed and tested during the development phase.

For some languages and subtasks, we changed the approach because with the extended dataset, we were able to achieve better results on some of the languages with a different approach. The parameters and models shown in appendix A.8 are what was finally used for the predictions submitted.

The F1-macro scores achieved in the test data set can be found in table 1 together with a comparison to the POLAR baseline. The table shows that our approach achieved better results especially in subtasks 2 and 3, which shows that language specific models outperform a standard approach applied to all languages.

The table 2 in the appendix shows the official rankings compared to the total number of submissions.

6 Discussion

Our results consistently demonstrate that language-specific pre-trained models outperform generic multilingual baselines, particularly for complex and low-resource languages. In order to get a good average result over all subtasks, we explored some options to improve the results per language and subtask to a certain degree.

Our goal was to provide methods that would provide a good average result in all languages and subtasks.

We developed a few patterns that we could apply to the training for all languages and that would work with different BERT and RoBERTa models. These models can be easily changed, so that we could find the optimal model for each of the 22 languages.

We also showed that some approaches did not work well, like a zero shot LLM approach on the binary polarization classification.

7 Future work

There are still a couple of approaches and experiments that could be done to see if this would improve the final results.

Since some of the languages belong to the same language families, we might be able to take advantage of this effect and try using it to train and finetune a model on multiple languages from the same family at once. An indicator that this approach might work well is shown in our selection of pretrained models. Often, we did select the same model for an entire family of languages. An example of this was the model chosen for most Indian languages (Khanuja et al., 2021).

Our LLM experiments used only small general-purpose models. Larger or task-adapted models, combined with more principled few-shot example selection, may show stronger results.

References

Abdulhady Abas Abdullah, Amir H. Gandomi, Tarik A Rashid, Seyedali Mirjalili, Laith Abualigah, Milena Živković, and Hadi Veisi. 2025. The role of orthographic consistency in multilingual embedding models for text classification in arabic-script languages. *Preprint*, arXiv:2507.18762.

Aimlab. 2021. XLM-RoBERTa Base Fine-tuned Urdu. https://huggingface.co/Aimlab/xlm-roberta-base-finetuned-urdu.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2020. Asymmetric loss for Multi-Label classification.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Jose Camacho-collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa Anke, Fangyu Liu, Eugenio Martínez Cámara, and 1 others. 2022. TweetNLP: Cutting-edge natural language processing for social media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–49, Abu Dhabi, UAE. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. *Preprint*, arXiv:2010.10906.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

deepset. 2019. German BERT. `https://huggingface.co/google-bert/bert-base-german-cased`.

Daryna Dementieva, Valeriia Khylenko, and Georg Groh. 2025. Cross-lingual text classification transfer: The case of Ukrainian. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1451–1464, Abu Dhabi, UAE. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021b. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Georg Hofmann and Annemarie Friedrich. 2025. Coling-UniA at GermEval 2025 shared task on candy speech detection: Retrieval augmented generation for identifying expressions of positive attitudes in German YouTube comments. In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops*, pages 404–410, Hannover, Germany. HsH Applied Academics.

Raviraj Joshi. 2022a. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Raviraj Joshi. 2022b. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages. *Preprint*, arXiv:2103.10730.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language.

Dariusz Kłeczek. 2020. PolBERT: Attacking Polish NLP tasks with transformers. `https://github.com/kldarek/polbert`. Blog post and model release.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Llama Team, AI @ Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783. Llama-3.1-8B-Instruct: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct.

Marzieh Farahani Mohammad Manthouri Mehrdad Farahani, Mohammad Gharachorloo. 2020. Parsbert: Transformer-based model for persian language understanding. *ArXiv*, abs/2005.12515.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *Preprint*, arXiv:2006.04884.

Usman Naseem, Robert Geislinger, Juan Ren, Sarah Kohail, Rudy Garrido Veliz, P Sam Sahil, Yiran Zhang, Marco Antonio Stranisci, Idris Abdulmumin, "Ozge Alacam, Cengiz Acar"urk, Aisha Jabr, Saba Anwar, Abinew Ali Ayele, Elena Tutubalina, Aung Kyaw Htet, Xintong Wang, Surendrabikram Thapa, Tanmoy Chakraborty, and 15 others. 2026a. SemEval-2026 task 9: Detecting multilingual, multicultural and multievent online polarization. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*. Association for Computational Linguistics.

Usman Naseem, Robert Geislinger, Juan Ren, Sarah Kohail, Rudy Garrido Veliz, P Sam Sahil, Yiran Zhang, Marco Antonio Stranisci, Idris Abdulmumin, Özge Alacam, Cengiz Acartürk, Aisha Jabr, Saba Anwar, Abinew Ali Ayele, Simona Frenda, Alessandra Teresa Cignarella, Elena Tutubalina, Oleg Rogov, Aung Kyaw Htet, and 24 others. 2026b. Polar: A benchmark for multilingual, multicultural, and multi-event online polarization. *Preprint*, arXiv:2505.20624.

NLP Town. 2019. Multilingual Sentiment BERT. https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment.

PKO Bank Polski. 2021. Polish RoBERTa. https://huggingface.co/PKOBP/polish-roberta-8k.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Arjun Sapkota. 2022. Nepali Sentiment BERT. https://huggingface.co/arsapkota/nepali-sentiment-bert.

Stefan Schweter. 2020a. BERTino - Italian BERT models. Https://huggingface.co/dbmdz/bert-base-italian-xxl-cased.

Stefan Schweter. 2020b. Berturk - bert models for turkish.

ssary. 2021. XLM-RoBERTa German Sentiment. https://huggingface.co/ssary/XLM-RoBERTa-German-sentiment.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *Preprint*, arXiv:2305.08377.

Zengzhi Wang, Qiming Xie, Yi Feng, Zixiang Ding, Zinong Yang, and Rui Xia. 2024. Is chatgpt a good sentiment analyzer? a preliminary study. *Preprint*, arXiv:2304.04339.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Seanghay Yath. 2022. XLM-RoBERTa Khmer Small. https://huggingface.co/seanghay/xlm-roberta-khmer-small.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning. *Preprint*, arXiv:2006.05987.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906, Mexico City, Mexico. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1298–1308.

Dmitry Zmitrovich, Alexander Abramov, Andrey Kalmykov, Maria Tikhonova, Ekaterina Taktasheva, Danil Astafurov, Mark Baushenko, Artem Snegirev, Tatiana Shavrina, Sergey Markov, Vladislav Mikhailov, and Alena Fenogenova. 2023. A family of pretrained transformer language models for russian. *Preprint*, arXiv:2309.10931.

A Appendix

A.1 Dataset analysis plots

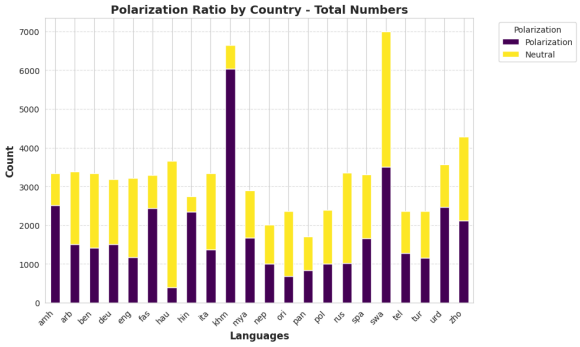

Figure 1: The ratio between polarizing and none polarizing statements per country.

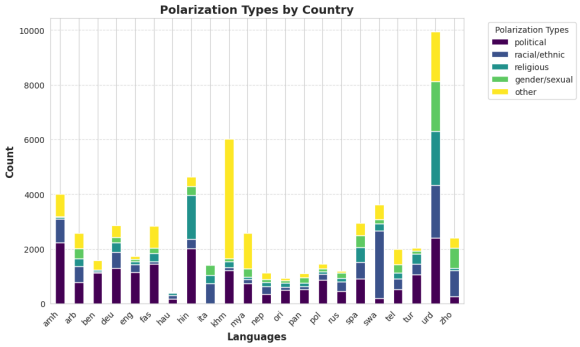

Figure 2: The polarization types per language. This can contain multiple labels on the same statements so the graph does not show a total absolute number.

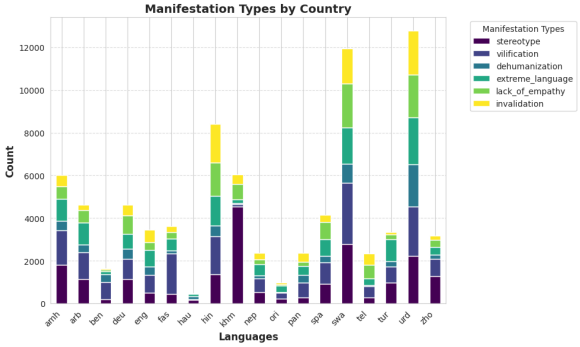

Figure 3: The manifestation types per language. This can contain multiple labels on the same statements so the graph does not show a total absolute number.

A.2 Development Set Results

		Subtask 1			Subtask 2			Subtask 3		
Language	Name	Best Score	Our Score	Diff	Best Score	Our Score	Diff	Best Score	Our Score	Diff
amh	Amharic	0.7847	0.7831	0.0016	0.5145	0.4493	0.0652	0.5475	0.4988	0.0487
arb	Arabic	0.8391	0.7800	0.0591	0.6255	0.6166	0.0089	0.6374	0.5780	0.0594
ben	Bengali	0.8948	0.8505	0.0443	0.473	0.4220	0.0510	0.2493	0.2147	0.0346
deu	German	0.7924	0.7734	0.0190	0.5624	0.4756	0.0868	0.5291	0.4438	0.0853
eng	English	0.8902	0.8276	0.0626	0.5026	0.4339	0.0687	0.5188	0.4890	0.0298
fas	Persian	0.9105	0.8545	0.0560	0.6393	0.5611	0.0782	0.4057	0.4140	-0.0083
hau	Hausa	0.8488	0.7847	0.0641	0.3477	0.3200	0.0277	0.3134	0.1693	0.1441
hin	Hindi	0.8587	0.8375	0.0212	0.8418	0.8269	0.0149	0.7965	0.7688	0.0277
ita	Italian	0.7036	0.6785	0.0251	0.4343	0.3903	0.0440			
khm	Khmer	0.6964	0.7027	-0.0063	0.7725	0.6901	0.0824	0.4604	0.4030	0.0574
mya	Burmese	0.8878	0.8526	0.0352	0.6623	0.6142	0.0481			
nep	Nepali	0.9000	0.8394	0.0606	0.7934	0.7635	0.0299	0.7025	0.6177	0.0848
ori	Oriya, Odia	0.8531	0.8121	0.0410	0.6311	0.6099	0.0212	0.3228	0.2965	0.0263
pan	Punjabi	0.8796	0.8193	0.0603						
pol	Polish	0.8323	0.7758	0.0565	0.6091	0.5400	0.0691			
rus	Russian	0.8349	0.7426	0.0923	0.6061	0.5214	0.0847			
spa	Spanish	0.7512	0.7145	0.0367	0.6442	0.6199	0.0243	0.4971	0.4845	0.0126
swa	Swahili	0.8480	0.8509	-0.0029	0.5548	0.4994	0.0554	0.5797	0.5127	0.0670
tel	Telugu	0.9236	0.8729	0.0507	0.4887	0.4919	-0.0032	0.3850	0.3813	0.0037
tur	Turkish	0.8608	0.7823	0.0785	0.6809	0.6387	0.0422	0.4699	0.4742	-0.0043
urd	Urdu	0.8094	0.7330	0.0764	0.8159	0.7392	0.0767	0.8085	0.7679	0.0406
zho	Chinese	0.9486	0.8925	0.0561	0.811	0.7598	0.0512	0.6483	0.6765	-0.0282
	Average	0.8431	0.7982	0.0449	0.6196	0.5707	0.0489	0.5219	0.4818	0.0401

Figure 4: Development results for our approach compared to the best observed results shown on Codabench at the end of December 2025. Red highlights a larger difference to the best result to green, where there is almost no difference. Negative values means that we surpassed the previous published results.

A.3 Official Test Set Rankings

Language	Subtask 1	Subtask 2	Subtask 3
amh	11 (30)	11 (21)	3 (15)
arb	22 (33)	5 (23)	6 (15)
ben	16 (37)	7 (26)	4 (17)
deu	23 (33)	22 (24)	12 (15)
eng	6 (44)	25 (29)	11 (18)
fas	15 (32)	8 (22)	5 (15)
hau	7 (31)	12 (22)	6 (16)
hin	25 (35)	8 (25)	4 (17)
ita	7 (32)	19 (22)	–
khm	4 (31)	11 (21)	3 (15)
mya	27 (30)	16 (21)	–
nep	29 (33)	12 (23)	8 (16)
ori	24 (33)	2 (23)	7 (17)
pan	9 (33)	4 (20)	3 (16)
pol	11 (32)	6 (23)	–
rus	25 (31)	12 (23)	–
spa	27 (36)	18 (25)	7 (17)
swa	5 (31)	12 (22)	3 (16)
tel	9 (33)	5 (23)	5 (17)
tur	26 (31)	10 (22)	7 (15)
urd	31 (35)	12 (25)	8 (17)
zho	18 (33)	12 (23)	8 (17)
Average	17.14	11.32	6.11

Table 2: Official test set ranking results. Values represent the rank achieved by the described approaches. Number in brackets show the total number of submissions the language and subtask. Not all languages were provided for subtask 3.

A.4 BERT and RoBERTa model references

- FacebookAI/xlm-roberta-large (Conneau et al., 2019)
- FacebookAI/xlm-roberta-base (Conneau et al., 2019)
- FacebookAI/roberta-large (Liu et al., 2019)
- Davlan/xlm-roberta-base-finetuned-amharic (Alabi et al., 2022)
- Davlan/bert-base-multilingual-cased-finetuned-hausa (Alabi et al., 2022)
- Davlan/bert-base-multilingual-cased-finetuned-swahili (Alabi et al., 2022)
- asafaya/bert-base-arabic (Safaya et al., 2020)
- asafaya/bert-large-arabic (Safaya et al., 2020)
- l3cube-pune/bengali-bert (Joshi, 2022a)
- deepset/gbert-base (Chan et al., 2020)
- google-bert/bert-large-uncased (Devlin et al., 2018)
- google-bert/bert-base-german-cased (deepset, 2019) based on (Devlin et al., 2018)
- ssary/XLM-RoBERTa-German-sentiment (ssary, 2021) based on (Conneau et al., 2019)
- microsoft/deberta-v3-large (He et al., 2021a)
- cardiffnlp/twitter-roberta-base-sentiment-latest (Camacho-collados et al., 2022)
- HooshvareLab/bert-base-parsbert-uncased (Mehrdad Farahani, 2020)
- HooshvareLab/bert-fa-base-uncased-clf-persiannews (Mehrdad Farahani, 2020)
- dbmdz/bert-base-italian-xxl-cased (Schweter, 2020a)
- dbmdz/bert-base-turkish-cased (Schweter, 2020b)
- seanghay/xlm-roberta-khmer-small (Yath, 2022) based on (Conneau et al., 2019)
- nlptown/bert-base-multilingual-uncased-sentiment (NLP Town, 2019)
- arsapkota/nepali-sentiment-bert (Sapkota, 2022)
- google/muril-base-cased (Khanuja et al., 2021)
- PKOBP/polish-roberta-8k (PKO Bank Polski, 2021) based on (Liu et al., 2019)
- dkleczek/bert-base-polish-uncased-v1 (Kłeczek, 2020)
- ai-forever/ruBert-base (Zmitrovich et al., 2023)
- DeepPavlov/rubert-base-cased (Kuratov and Arkhipov, 2019)
- dccuchile/bert-base-spanish-wwm-uncased (Cañete et al., 2020)
- Aimlab/xlm-roberta-base-finetuned-urdu (Aimlab, 2021) based on (Conneau et al., 2019)
- hfl/chinese-macbert-large (Cui et al., 2020)
- hfl/chinese-macbert-base (Cui et al., 2020)
- microsoft/mdeberta-v3-base (He et al., 2021b)
- l3cube-pune/telugu-bert (Joshi, 2022b)
- csebuetnlp/banglabert (Bhattacharjee et al., 2022)
- deepset/gelectra-base (Chan et al., 2020)

A.5 LLM model references

- Qwen/Qwen2.5-7B-Instruct (Yang et al., 2024)
- meta-llama/Llama-3.1-8B-Instruct (Llama Team, AI @ Meta, 2024)

A.6 Subtask 1 - LLM prompt template

Prompt A.1: Binary classification prompt for Subtask 1

You are an export for text annotations for hate and polarization evaluation.
The used language is `{language}`
Task:
For the given statement, evaluate if it contains hateful, abusive, or polarizing content.
Classes:
YES for hateful, abusive, or polarizing content
NO for neutral content
Here are samples of polarizing or hateful content:
`{positive_examples}`
And these are samples of neutral content:
`{negative_examples}`
Now classify the new statement.
Instructions:

- read the post carefully
- decide which class is more appropriate
- answer with YES or NO
- do not include any additional explanations

New statement:
`{statement}`
Answer YES or NO only

A.7 Subtask 2/3 - LLM prompt template

Prompt A.2: Few-Shot Multi-label classification prompt for the engish language

You are an expert text classifier specialized in identifying polarization in text. CRITICAL: Only mark text as polarized if you are confident in your prediction. When in doubt, choose "None". Multiple labels can be chosen for a text. **Task: Step-by-Step Polarization Analysis**
Analyze the text using this EXACT process:
STEP 1: List key phrases that might indicate polarization
STEP 2: Check for us-vs-them framing or group hostility (yes/no)
STEP 3: Determine if polarized (true/false)
STEP 4: If polarized, match to specific labels from the definitions below (Multiple labels can be assigned to a single text, the task is a Multilabel Classification)
Labels
The available Labels are: `{taxonomy}`
Examples:
Example 1 (NOT polarized):
Input: "Still playing with this. I am now following Rachel Maddow from msnbc"

```
<json>
{
"step1_key_phrases": ["following", "msnbc"],
"step2_hostility": "no",
"step3_is_polarized": false,
"labels": ["none"],
"final_reasoning": "Personal action. No divisive language or group hostility."
}
</json>
```

Example 2 (POLARIZED - multiple labels):
Input: "Although obviously letting Azovstal prisoners go isn't helping denazification."

```
<json>
{
"step1_key_phrases": ["Azovstal", "denazification", "prisoners"],
"step2_hostility": "yes",
"step3_is_polarized": true,
"labels": ["political", "racial/ethnic"],
"final_reasoning": "Political: Russia-Ukraine conflict rhetoric. Racial/ethnic:
'denazification' frames ethnic/national tensions."
}
</json>
```

Input Text:

```
"{text}"
```

Your Analysis (output ONLY the JSON):

```
<json>
{
"step1_key_phrases": [],
"step2_hostility": "yes/no",
"step3_is_polarized": True/False,
"labels": [],
"final_reasoning": ""
}
</json>
```

A.8 Subtasks - strategy and models

A.8.1 Amharic (amh)

	Subtask		
	1	**2**	**3**
Strategy	RoBERTa		
Model	xlm-roberta-base-finetuned-amharic (Alabi et al., 2022)		
Split	5F-CV	5F-CV	90/10
Epochs	5 each	5 each	20
Weighted	yes	yes	yes

A.8.2 Arabic (arb)

Subtask 1	
Strategy	BERT
Model	asafaya/bert-base-arabic (Safaya et al., 2020)
Split	90/10
Epochs	20
Weighted	no

	Subtask 2	**Subtask 3**
Strategy	BERT	
Model	asafaya/bert-large-arabic (Safaya et al., 2020)	
Split	5F-CV	90/10
Epochs	5 each	20
Weighted	yes	yes

A.8.3 Bengali (ben)

Subtask 1	
Strategy	BERT
Model	l3cube-pune/bengali-bert (Joshi, 2022a)
Split	90/10
Epochs	20
Weighted	yes

Subtask 2	
Strategy	BERT
Model	google/muril-large-cased (Khanuja et al., 2021)
Split	90/10
Epochs	20
Weighted	yes

Subtask 3	
Strategy	BERT
Model	csebuetnlp/banglabert (Bhattacharjee et al., 2022)
Split	90/10
Epochs	20
Weighted	yes

A.8.4 German (deu)

Subtask 1	
Strategy	multiple (Ro)BERT(a) majority vote
Models	deepset/gbert-base (Chan et al., 2020) bert-base-german-cased (deepset, 2019) RoBERTa-German-sentiment (ssary, 2021)
Split	90/10
Epochs	20 each
Weighted	no

Subtask 2	
Strategy	RoBERTa
Model	RoBERTa-German-sentiment (ssary, 2021)
Split	90/10
Epochs	20
Weighted	yes
Individual	yes

Subtask 3	
Strategy	BERT
Model	deepset/gelectra-base (Chan et al., 2020)
Split	90/10
Epochs	20
Weighted	yes

A.8.5 English (eng)

Subtask 1	
Strategy	multiple (Ro)BERT(a) and LLMs, majority vote
Models	twitter-roberta-base -sentiment-latest (Camacho-collados et al., 2022) bert-large-uncased (Devlin et al., 2018) roberta-large (Liu et al., 2019)
LLMs	Qwen2.5-7B-Instruct (Yang et al., 2024) Llama-3.1-8B-Instruct (Llama Team, AI @ Meta, 2024)
Split	90/10
Epochs	20 each
Weighted	yes

Subtask 2	
Strategy	RoBERTa
Model	twitter-roberta-base-sentiment-latest (Camacho-collados et al., 2022)
Split	90/10
Epochs	20
Weighted	yes
Individual	yes

Subtask 3	
Strategy	RoBERTa
Model	twitter-roberta-base-sentiment-latest (Camacho-collados et al., 2022)
Split	90/10
Epochs	20
Weighted	yes

A.8.6 Persian (fas)

Subtask 1	
Strategy	BERT
Model	bert-fa-base-uncased-clf-persiannews (Mehrdad Farahani, 2020)
Split	90/10
Epochs	20
Weighted	no

Subtask 2 and 3	
Strategy	BERT
Model	bert-base-parsbert-uncased (Mehrdad Farahani, 2020)
Split	90/10
Epochs	20
Weighted	yes

A.8.7 Hausa (hau)

	Subtask		
	1	**2**	**3**
Strategy	BERT		
Model	bert-base-multilingual -cased-finetuned-hausa (Alabi et al., 2022)		
Split	90/10		
Epochs	20	20	40
Weighted	yes		
Individual	-	yes	no

A.8.8 Hindi (hin)

	Subtask		
	1	**2**	**3**
Strategy	BERT		
Model	muril-base-cased (Khanuja et al., 2021)		
Split	90/10		
Epochs	20		
Weighted	no	yes	yes

A.8.9 Italian (ita)

	Subtask	
	1	**2**
Strategy	BERT	
Model	bert-base-italian	
	-xxl-cased	-italian-uncased
	(Schweter, 2020a)	
Split	90/10	
Epochs	20	
Weighted	yes	

A.8.10 Khmer (khm)

	Subtask		
	1	**2**	**3**
Strategy	RoBERTa		
Model	xlm-roberta-khmer-small (Yath, 2022)		
Split	5F-CV		90/10
Epochs	5 each		20
Weighted	no	yes	yes

A.8.11 Burmese (mya)

	Subtask		
	1	**2**	**3**
Strategy	BERT	RoBERTa	
Model	bert-base-*-uncased-sentiment [3] (NLP Town, 2019)	xlm-roberta-base (Khanuja et al., 2021)	
Split	90/10		
Epochs	20		
Weighted	yes	yes	yes

A.8.12 Nepali (nep)

Subtask		
	1	**3**
Strategy	BERT	
Model	arsapkota/nepali-sentiment-bert (Sapkota, 2022)	
Split	90/10	
Epochs	20	
Weighted	yes	yes

Subtask 2	
Strategy	BERT
Model	google/muril-base-cased (Khanuja et al., 2021)
Split	90/10
Epochs	20
Weighted	yes

A.8.13 Oriya (ori)

	Subtask		
	1	**2**	**3**
Strategy	BERT		
Model	muril-*-cased		
	base	large	base
	(Khanuja et al., 2021)		
Split	5F-CV	90/10	
Epochs	5 each	20	
Weighted	yes	yes	yes

[3] nlptown/bert-base-multilingual-uncased-sentiment

A.8.14 Punjabi (pan)

	Subtask	
	1	**2**
Strategy	BERT	
Model	muril-*-cased	
	large	base
	(Khanuja et al., 2021)	
Split	90/10	5F-CV
Epochs	10	5 each
Weighted	yes	yes

A.8.15 Polish (pol)

Subtask 1	
Strategy	multiple RoBERTa plus LLMs majority vote
Models	xlm-roberta-large (Conneau et al., 2019) roberta-large (Liu et al., 2019) PKOBP/polish-roberta-8k (PKO Bank Polski, 2021)
LLM	Qwen2.5-7B-Instruct (Yang et al., 2024) Llama-3.1-8B-Instruct (Llama Team, AI @ Meta, 2024)
Split	90/10 (RoBERTa)
Epochs	20 each (RoBERTa)
Weighted	no

Subtask 2	
Strategy	RoBERTa
Model	PKOBP/polish-roberta-8k (PKO Bank Polski, 2021)
Split	90/10
Epochs	20
Weighted	yes

A.8.16 Russian (rus)

Subtask		
	1	**2**
Strategy	BERT	
Model	rubert-base-cased (Kuratov and Arkhipov, 2019)	ruBert-base (Zmitrovich et al., 2023)
Split	5F-CV	90/10
Epochs	5 each	20
Weighted	no	yes

A.8.17 Spanish (spa)

Subtask		
	1	3
Strategy	BERT	
Model	dccuchile/ bert-base-spanish -wwm-uncased (Cañete et al., 2020)	
Split	5F-CV	90/10
Epochs	5 each	20
Weighted	no	yes

Subtask 2	
Strategy	RoBERTa
Model	xlm-roberta-base (Conneau et al., 2019)
Split	90/10
Epochs	20
Weighted	yes

A.8.18 Swahili (swa)

	Subtask		
	1	2	3
Strategy	BERT		
Model	bert-base-multilingual -cased-finetuned-swahili		
	(Alabi et al., 2022)		
Split	90/10		
Epochs	20		
Weighted	no	yes	yes

A.8.19 Telugu (tel)

Subtask		
	1	3
Strategy	BERT	
Model	google/muril-base-cased (Khanuja et al., 2021)	
Split	5F-CV	
Epochs	5 each	
Weighted	no	yes

Subtask 2	
Strategy	BERT
Model	l3cube-pune/telugu-bert (Joshi, 2022b)
Split	90/10
Epochs	20
Weighted	yes

A.8.20 Turkish (tur)

	Subtask		
	1	2	3
Strategy	BERT		
Model	bert-base-turkish-cased (Schweter, 2020b)		
Split	90/10	5F-CV	90/10
Epochs	20	5 each	20
Weighted	yes		

A.8.21 Urdu (urd)

	Subtask		
	1	2	3
Strategy	(Ro)BERT(a)		
Model	finetuned-urdu [4] (Aimlab, 2021)	mdeberta-v3-base (He et al., 2021b)	muril-base-cased (Khanuja et al., 2021)
Split	90/10		
Epochs	20		
Weighted	no	yes	yes

A.8.22 Chinese (zho)

	Subtask		
	1	2	3
Strategy	BERT		
Model	hfl/chinese-macbert-*		
	large	base	large
	(Cui et al., 2020)		
Split	5F-CV	90/10	
Epochs	5 each	20	
Weighted	no	yes	yes

[4] Aimlab/xlm-roberta-base-finetuned-urdu