Groningen team D at SemEval-2024 Task 8: Exploring data generation and a combined model for fine-tuning LLMs for Multidomain Machine-Generated Text Detection

Thijs Brekhof, Xuanyi Liu, Joris Ruitenbeek, Niels Top, Yuwen Zhou
University of Groningen
t.j.brekhof@student.rug.nl, x.liu.69@student.rug.nl,
j.e.j.ruitenbeek@student.rug.nl, n.top@student.rug.nl, y.zhou.74@student.rug.nl

Abstract

In this system description, we report our process and the systems that we created for the subtasks A monolingual, A multilingual, and B for the SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection. (Wang et al., 2024) This shared task aims at discriminating between machine-generated text and human-written text. Subtask A focuses on detecting if a text is machine-generated or human-written both in a monolingual and a multilingual setting. Subtask B also focuses on detecting if a text is human-written or machine-generated, though it takes it one step further by also requiring the detection of the correct language model used for generating the text. For the monolingual aspects of this task, our approach is centered around fine-tuning a deberta-v3-large LM. For the multilingual setting, we created a combined model utilizing different monolingual models and a language identification tool to classify each text. We also experiment with the generation of extra training data. Our results show that the generation of extra data aids our models and leads to an increase in accuracy.

1 Introduction

The SemEval-2024 shared task focuses on multi-generator, multidomain, and multilingual black-box machine-generated text detection. The shared task is split into three different subtasks. Each subtask is monolingual except for the first subtask, which has a monolingual (English) and a multilingual track. The languages covered in this shared task include English, Chinese, Russian, Urdu, Indonesian, Italian, German, and Arabic.

This paper presents our the systems that we created for the shared task. The paper provides an overview of our research strategies and results for subtasks A and B.

Subtask A focuses on the detection of machine-versus human-written text, we differentiate between mono- and multilingual data. Our approach involves fine-tuning LLMs, DeBERTa-v3 (large) in particular. We experimented with different parameters for the model, searching for the best performance possible.

Subtask B extends the challenge presented in subtask A, we now attempt to recognize the specific language model used for text generation. We do this in addition to distinguishing between human and machine-generated text. We again use DeBERTa-v3 (large) to classify the data. To optimize model accuracy, we fine-tune hyperparameters.

Additionally, we generate extra Wikipedia articles to further expand the training data. We hypothesize that extra data will lead to better model performance, and thus better applicability to real-world applications. Our research focuses on finding both the best possible language model settings to recognize machine- and human-written text and distinguish between different language generation models. Our code and the additionally generated data can be found on Github[1]

2 Related work

Previous research has been done on the topic of automatically discriminating between human-text and machine-generated text (Chichirau et al., 2023), where DeBERTa (v3) (He et al., 2021) is utilized as a target-only classifier. The model can distinguish machine translations well when tested on the test set after training on texts generated from different source languages and different ma-

[1] https://github.com/thijsbrekhof1/RUG-D-at-SemEval2024-task8

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 391–398
June 20-21, 2024 ©2024 Association for Computational Linguistics

	Train	Dev	Test
Subtask A-Mono	119757	5000	34272
Subtask A-Multi	172417	4000	42378
en	136589	0	28200
ar	0	1000	2103
ru	0	2000	0
zh	11934	0	0
id	5995	0	0
ur	5899	0	0
bg	12000	0	0
de	0	1000	6000
it	0	0	6075
Subtask B	71027	3000	18000

Table 1: Statistics of train, dev, and test sets provided by organizers

chine translation systems. They found that both the monolingual and multilingual DeBERTa models outperformed other LLMs that they evaluated.

Langid.py (Lui and Baldwin, 2012) is a supervised language identification tool trained using a naive Bayes classifier. Langid.py has the following advantages: fast, usable off-the-shelf, unaffected by domain-specific features (e.g. HTML, XML, markdown), single file with minimal dependencies, and flexible interface. Langid.py was applied in our system to identify the multi-language training set of subtask A and we found that it can identify languages with very high accuracy.

3 Data

The dataset provided by the shared task creators originates from the benchmark M4 (Wang et al., 2023). M4 is a comprehensive dataset encompassing machine-generated text from diverse generators, domains, and languages. M4 focuses on the development of automated systems for detecting machine-generated text and identifying potential abuse.

The dataset comprises text samples sourced from various platforms, including Wikipedia, Reddit, WikiHow, PeerRead, Arxiv, Chinese QA, Urdu News, Russian RuATD, Indonesian News, and Arabic Wikipedia. It spans multiple languages and domains, presenting a rich and diverse collection of machine-generated text for analysis and classification.

Table 1 presents the statistics of the dataset, including the number of samples in the train, dev, and test sets for subtasks A and B. For subtask A, both monolingual (subtask A-Mono) and multilingual (subtask A-Multi) tracks are included, with train, dev, and test set sizes specified for each language. Subtask B involves multi-way classification of machine-generated text and includes corresponding train, dev, and test set sizes.

4 System overview

This section presents an overview of the methods we employed for subtask A, both the monolingual and multilingual data setting, as well as subtask B. We follow previous work on a similar topic (Chichirau et al., 2023), by fine-tuning LLMs, predominantly DeBERTa (He et al., 2021), on this task. We were further stimulated to explore this model specifically, as DeBERTa is developed as an improvement over the RoBERTa model (Liu et al., 2019), the latter being employed by the task organizers as a baseline. Specifically, we looked at using both the base and large variants of deberta-v3, as this improved version of DeBERTa is reported to significantly outperform previous iterations on numerous tasks.

As the goal of this task was to create systems that can discriminate between human-written and machine-generated text regardless of generator, textual domain, or language, we opted not to preprocess our data any further than what the task organizers already did. This will keep our data as close to instances that can be encountered in real-world scenarios as possible. We fine-tuned these pre-trained language models using the Transformers library from Huggingface (Wolf et al., 2020).

4.1 Subtask A: Monolingual

For the monolingual track of subtask A, we evaluated the performance of the base (86M parameters) and large (304M parameters) variants of DeBERTa. We tested out numerous combinations of hyperparameters such as learning rate, batch size, maximum input sequence length, and epochs to found out which model would perform best. The large DeBERTa model emerged superior over the base model, ostensibly due to its larger model size.

For this track of the task, we also experimented with generating additional training data. The goal for this subtask is for our model to differentiate between human-written and machine-generated text, regardless of what generative model was used to obtain data. We were inclined to experiment with additional data generation by a model different from the ones already present in the provided base dataset, as this should allow our model to generalize better across generators and not learn only about those present in the base dataset. For potential real-world applications, this would be especially interesting to experiment with, as in such scenarios there would be no prior indication of what model could be used to generate such texts.

We employed Llama 2 (Touvron et al., 2023) to generate additional articles in the style of Wikipedia and manually skimmed through the generated texts to see if they were on a comparable level to the data provided by the task organizer. Subsequently, we took the hyperparameter configurations of our best-performing model trained on only base data and trained a new model using the same configurations on a combination of the base data and our additionally generated articles. The selection of Wikipedia as our domain of focus is based on its comprehensible documentation and the strong performance demonstrated by LLama 2 in generating texts within this specific domain.

4.2 Subtask A: Multilingual

Different from the monolingual strategy, we created a combined model for this subtrack. We explored a way to use separate monolingual models for different languages after determining the language of each text. After discovering that there was no data in the same language both in the original train and dev set (see Table 1, we decided to merge the two data sets and extract each language separately for analysis. We embarked on a language-specific modeling approach, recognizing the importance of selecting models optimized for each language's unique characteristics.

To determine the most suitable approach for each language, we compared the performance of multilingual DeBERTa with specific monolingual models. We employed a 10-fold cross-validation approach within each language, evaluating models based on accuracy and standard deviation. The best-performing model for each language was selected for further evaluation.

Upon completion of the cross-validation procedure, we selected the model that exhibited the highest performance on the development set for each language. The selected models were then applied to the test set for final evaluation, encompassing the full spectrum of languages represented in the dataset. To handle the multilingual nature of the test set effectively, we employed the language identification tool Langid, to discern the language of each text sample, which enabled us to tailor model predictions to the specific linguistic context of each sample.

Notably, we also employed Llama 2 to generate additional training data for each language. We utilized a 10-fold cross-validation process to assess the impact of additional training data on model performance across different languages and only kept those that improved the results.

4.2.1 LangID

In our multilingual subtask A experiment, we proposed the idea of using specific language models per language instead of a single model for each of the languages. Our motivation was that this approach could improve the accuracy of discriminating between machine-generated text and human-written texts better than a single multilingual model could. To achieve this goal, we employed LangID to enable language-specific modeling. After merging the train and dev sets and extracting samples for each language separately, we utilized LangID to determine the language of each text sample in the test set and employed MDeBERTa-v3-base for languages that were not in the train or dev sets and could not be recognized by LangID. Thus, we were able to effectively handle the multilingual nature of the task.

4.3 Subtask B

For this subtask, we, similarly to our approach for subtask A, compared the performance of the base and large variant of DeBERTa. By testing out different values for epochs, learning rate, maximum input sequence length, and batch size, we obtained the hyperparameter configurations of our best-performing model. The large variant of DeBERTa once again outperformed the base version.

We opted not to use additionally generated data for this subtask. The goal of subtask B is to determine not only if the text is human-written or machine-generated but also what generative model was used to do so. This would make generating data by models outside of the already provided list of models in the base dataset futile.

4.4 Generating data

While we realize that it is not allowed to add additional data for the shared task we see generating it as a real-world contribution that can also easily be done by others. We generated our own extra training data with the use of Llama 2 (Touvron et al., 2023). Starting off, we wanted to exploit the largest model available, because this should offer the best performance in data generation. However, due to limited resources, we opted to utilize the 7 billion parameter version.

We focussed our generation endeavors on languages that were already in the dataset but were highly underrepresented. These included Russian, Arabic, German, and Indonesian. For each of these languages, we extended the dataset so that each of these languages had a total of 30,000 samples. Notably, for every sample generated by the model, we also included a human-written counterpart in the dataset. By doing this, we aimed to maintain a balance between computer- and human-written data in the training and development sets.

To match the already generated Wikipedia articles in the dataset, we adopted a similar method to the original M4 dataset, as outlined by (Wang et al., 2023). Using the Wikipedia dataset available on HuggingFace (Wikimedia-Foundation, 2023), we randomly selected articles with a minimum length of 1,000 characters. Subsequently, we prompted Llama 2 to generate Wikipedia articles based on provided titles. As an extra criterion, we told the model that the resulting articles should contain at least 250 words, as this was also the criteria used in the original paper (Wang et al., 2023). This approach enabled us to enrich our dataset across multiple languages, with the purpose of increasing the performance of our models.

5 Experimental setup

5.1 Datasets and Evaluation Metrics

For both subtask A's monolingual part and subtask B, we utilized standard data splits: train, dev, and test sets. The train set was employed for model training, the dev set for monitoring performance and hyperparameter tuning, and the test set for final evaluation. Accuracy is the main evaluation metric to assess model performance in each task.

For multilingual subtask A, we adopted a different strategy, as motivated in Section 4.2. We concatenated the train and dev sets, extracted samples for different languages, and employed separate models for each language. We utilized the 10-fold cross-validation approach within each language to select the most suitable model based on accuracy and standard deviation. The selected models from each language were then used to predict the test set.

5.2 Training Details

For monolingual subtask A, the final selected hyperparameters were as follows: batch size 2, gradient accumulation 64, learning rate 1e-5, three epochs, formatting style fp16, and an input length of 1024 tokens.

For multilingual subtask A, we employed uniform hyperparameters throughout the 10-fold cross-validation process within each language. These hyperparameters included a learning rate of 2e-5, three epochs, a formatting style of fp16, and an input length of 512 tokens.

For subtask B, the following hyperparameters were identified as optimal: batch size 4, gradient accumulation 32, learning rate 1e-5, three epochs, formatting style fp16, and an input length of 512 tokens.

All of our hyperparameter values were chosen after extensive experimentation on the dev set to optimize model performance. A full list of all the hyperparameter values that we experimented with regarding the monolingual subtasks can be found by referring to Appendix A. Regarding multilingual subtask A, specific model selection and results for each language can be seen in Table 6 of Appendix B.

Additionally, all training processes were conducted on several Nvidia A100 and V100 GPUs.

6 Results and Analysis

In this section, we show and analyze the results achieved for each of the subtasks. Table 2 shows the quantitative results we achieved when running our models on our dev set and the organizer's test set. Tables 4, 7 and 5 in the appendix show the accuracy across languages and the impact of the usage of extra data on each subtask. Besides that, we made a qualitative analysis to find out where we think our systems make the most mistakes.

6.1 Analysis

Our analysis showed us several noteworthy points. First, our monolingual models achieved significantly higher scores on the dev sets than on the test set, as can be seen in Table 2. A reason for this could be the introduction of texts created by LLMs

Subtask	Baseline	Dev	Test
A Monolingual	88.46%	87.80%	63.68%
A Multilingual	80.88%	65.90%	71.79%
B	74.60%	72.80%	61.50%

Table 2: Scores of each subtask in dev and test compared to the baseline.

that our system had not seen before. This shows us a risk our systems may lack robustness against different types of LLMs. Our multilingual system did perform better on test than on dev, however, which could be related to the different ratios of languages present in both datasets. e.g., more German texts were present in the test dataset than in the dev dataset, and our system is able to classify them effectively, which can be seen in Table 6.

Furthermore, our systems were unable to effectively detect human-written texts, in both the mono- and multilingual tasks, when classifying the test set. In subtask A monolingual, our system was able to get very impressive scores on all texts created by generative models, though it had a lackluster performance on human-written text. This might indicate our model's inclination to classify a text as machine-generated over human-written. Subtask B has a very similar distribution of predictions, the only notable exception being the obstacle of detecting texts written by Cohere.

Also noticeable was the performance of texts generated by the Llama 2 model. Both our models with- and without added data scored badly on these texts. What is interesting, is that the extra data added by us originates from Llama 2. A reason for this could be that we used the smaller, 7 billion parameter, version of Llama 2 due to performance and runtime issues.

We can see that both in the mono- and multilingual data setting of subtask A our model's performance had improved after training on our extra generated data. Although the increase in accuracy of the monolingual model was negligible, the multilingual model had a notable improvement in score. We propose that this stems from the absence of certain languages in the training set, which we were able to supplement with our extra data. Because of this, the monolingual models we employed in the multilingual setting were able to perform better.

7 Discussion/Conclusion

In conclusion, we think our participation in the shared task resulted in some valuable insights into the challenges of machine- versus human-written text. Despite our efforts, our systems unfortunately fell short of surpassing the baseline scores established by the task organizers.

Across the different subtasks, our models showed varying performance. For subtask A monolingual, our models achieved some promising results on the development set, with an accuracy of 87.80%. However, our model did not manage to generalize enough, leading to an accuracy of 63.68% on the test set.

For the multilingual part of subtask A, our model reached 65.90% on the development set. In this case, the model did manage to generalize the data, leading to an accuracy of 71.79% on the test set. However, this was still below our expectations, and the baseline accuracy of 80.88%.

In subtask B, our models struggled to identify the specific language model used for text generation accurately, with accuracies of 72.80% on the development, and 61.50% on the test set. Despite optimizing hyperparameters and training on both original and additional data, our models failed to outperform the baseline accuracy of 74.60%.

We think our analysis revealed several points for improvement. Our models tended to misclassify human-written text, indicating a potential bias towards machine-generated content. Furthermore, the models seemed unable to generalize, leading to worse performance on the test set for monolingual task A.

Moving forward, we think there are many improvements to be made. Future research could focus on using other model architectures or exploring other data augmentation techniques. Also, training the model in more languages could improve the performance of multilingual models. Of course, using larger pre-trained models could also lead to an easy increase in performance, although it does require significant resources. Lastly, our findings also show generating extra training data is essential for improving model performance. Therefore, a promising direction for future work is to explore new data sources and methods to create richer and higher-quality training data to further improve the performance and generalization ability of the model.

8 Acknowledgements

This submission has been carried out as part of the 2023-2024 edition of the master course Shared Task Information Science (LIX026M05) at the University of Groningen, taught by Lukas Edman and Antonio Toral. We want to express our gratitude to the university and to both of our lecturers for assisting us in this project.

References

[Abdaoui et al.2020] Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of mutlilingual bert. In *SustaiNLP / EMNLP*.

[Antoun et al.2020] Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

[Chan et al.2020] Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

[Chichirau et al.2023] Malina Chichirau, Rik van Noord, and Antonio Toral. 2023. Automatic discrimination of human and neural machine translation in multilingual scenarios. In Mary Nurminen, Judith Brenner, Maarit Koponen, Sirkku Latomaa, Mikhail Mikhailov, Frederike Schierl, Tharindu Ranasinghe, Eva Vanmassenhove, Sergi Alvarez Vidal, Nora Aranberri, Mara Nunziatini, Carla Parra Escartín, Mikel Forcada, Maja Popovic, Carolina Scarton, and Helena Moniz, editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 217–226, Tampere, Finland, June. European Association for Machine Translation.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

[Guo et al.2023] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arxiv:2301.07597*.

[He et al.2021] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

[Kuratov and Arkhipov2019] Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language.

[Liu et al.2019] 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.

[Lui and Baldwin2012] Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30.

[Touvron et al.2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

[Wang et al.2023] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

[Wang et al.2024] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico, June.

[Wikimedia-Foundation2023] Wikimedia-Foundation. 2023. Wikimedia downloads.

[Wolf et al.2020] 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 Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Scao, Sylvain Gugger, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. pages 38–45, 01.

A Hyperparameters

Hyperparameter	Value
Learning rate	1e-5, 2e-5, 5e-5, 5e-6
Batch size	16, 32, 64, 128
Epoch	1, 2, 3
Input length	512, 768, 1024

Table 3: The full hyperparameter search space employed for our DeBERTa-v3-large model in both subtask A monolingual and subtask B.

B Detailed scores

Data	**Model**	**Accuracy**
A Monolingual - Original Data		
	Overall	63.61% (± 2.60E-3)
	Human	23.56% (± 3.33E-3)
	GPT4	99.77% (± 8.81E-4)
	Cohere	100.0% (0)
	ChatGPT	100.0% (0)
	Bloomz	99.1% (± 1.72E-3)
	Dolly	100.0% (0)
	Davinci	99.97% (± 3.33E-4)
A Monolingual - Added Data		
	Overall	63.68% (± 2.60E-03)
	Human	24.23% (± 3.36E-03)
	GPT4	99.9% (± 5.77E-04)
	Cohere	100.0% (0)
	ChatGPT	100.0% (0)
	Bloomz	96.2% (± 3.49E-03)
	Dolly	100.0% (0)
	Davinci	100.0% (0)

Table 4: Accuracy scores on the test set for subtask A Monolingual with original and added data.

Model	**Accuracy**
Overall	61.54% (± 3.63E-03)
Human	13.53% (± 1.37E-03)
Bloomz	99.43% (± 6.25E-03)
Dolly	86.1% (± 6.32E-03)
ChatGPT	99.93% (± 4.71E-04)
Cohere	1.23% (± 2.02E-03)
Davinci	69.0% (± 8.44E-03)

Table 5: Accuracy scores for subtask B on the test set.

Lang.	Model	Accuracy	Reference
en	deberta-v3-base	**95.9% ± 5.82E-3**	(He et al., 2021)
en	mdeberta-v3-base	95.6% ± 5.94E-3	(He et al., 2021)
ar	bert-base-arabert	**94.0% ± 4.10E-2**	(Antoun et al., 2020)
ar	mdeberta-v3-base	90.5% ± 4.46E-2	(He et al., 2021)
ru	rubert-base-cased	98.7% ± 8.12E-3	(Kuratov and Arkhipov, 2019)
ru	mdeberta-v3-base	**98.8% ± 6.00E-3**	(He et al., 2021)
zh	chatgpt-detector-roberta-chinese	**97.6% ± 5.64E-3**	(Guo et al., 2023)
zh	bert-base-chinese	96.8% ± 1.04E-2	(Devlin et al., 2019)
zh	mdeberta-v3-base	96.9% ± 1.37E-2	(He et al., 2021)
id	bert-base-indonesian-522M	**99.4% ± 4.68E-3**	
id	mdeberta-v3-base	98.8% ± 7.85E-3	(He et al., 2021)
ur	mdeberta-v3-base	**99.98% ± 5.08E-4**	(He et al., 2021)
bg	bert-base-en-bg-cased	97.2% ± 6.29E-3	(Abdaoui et al., 2020)
bg	mdeberta-v3-base	**99.3% ± 4.99E-3**	(He et al., 2021)
de	bert-base-german-cased	92.9% ± 3.53E-2	(Chan et al., 2020)
de	gbert-base	**93.8% ± 1.99E-2**	(Chan et al., 2020)
de	mdeberta-v3-base	91.0% ± 4.4E-2	(He et al., 2021)

Table 6: The accuracy and standard deviation of different models in each language under 10-fold cross validation. The best-performing models (in bold) were utilized in our combined model for multilingual subtask A. We only employed one (multilingual) model for Urdu, as we could not find any monolingual models trained on that language.

Data	Model	Accuracy	Language	Accuracy
A Multilingual - Original data				
	Overall	70.11% (± 2.22E-03)	English	72.32% (± 2.66E-03)
	Human	40.89% (± 4.28E-03)	German	84.45% (± 4.68E-03)
	ChatGPT	83.91% (± 3.51E-03)	Arabic	57.73% (± 1.08E-02)
	Bloomz	100.0% (0)	Italian	50.01% (± 6.42E-03)
	Davinci	99.9% (± 5.77E-04)		
	Llama 2	50.01% (± 6.42E-03)		
	Dolly	99.93% (± 4.71E-04)		
	Cohere	100.0% (0)		
	Jais-30b	61.29% (± 3.91E-02)		
A Multilingual - Added data				
	Overall	71.79% (± 2.19E-03)	English	72.32% (± 2.66E-03)
	Human	40.89% (± 4.28E-03)	German	90.92% (± 3.71E-03)
	ChatGPT	90.14% (± 2.85E-03)	Arabic	73.13% (± 9.67E-03)
	Bloomz	100.0% (0)	Italian	50.01% (± 6.42E-03)
	Davinci	99.9% (± 5.77E-04)		
	Llama 2	50.01% (± 6.42E-03)		
	Dolly	99.97% (± 3.33E-04)		
	Cohere	100.0% (0)		
	Jais-30b	80.0% (± 3.21E-02)		

Table 7: Accuracy scores and language-specific accuracies on the test set for subtask A Multilingual with original and added data.