NewbieML at SemEval-2024 Task 8: Ensemble Approach for Multidomain Machine-Generated Text Detection

Bao Tran and **Nhi Tran**
Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam
{bao.tran2003, nhi.tranluongyen}@hcmut.edu.vn

Abstract

Large Language Models (LLMs) are becoming popular and easily accessible, leading to a large growth of machine-generated content over various channels. Along with this popularity, the potential misuse is also a challenge for us. In this paper, we use subtask A monolingual dataset with comparative study between some machine learning model with feature extraction and develop an ensemble method for our system. Our system achieved 84.31% accuracy score in the test set, ranked 36th of 137 participants. Our code is available at: https://github.com/baoivy/SemEval-Task8

1 Introduction

In our current time, large language models (LLMs), such as ChatGPT (Ouyang et al., 2022), GPT-4[1], LLaMA (Touvron et al., 2023) and BLOOMz (Muennighoff et al., 2023) can be easily observed to be becoming increasingly prevalent from diverse forms ranging of news, multimedia to education. How outstandingly LLMs answer to user's problems makes them appealing for automatic missions as well as diminishing human labor in many scenarios. Nevertheless, this also unexpectedly leads to problems with regard to human's misuses, spreading misinformation and causing disruptions in the education system in particular. Therefore, it is necessary to develop systems that can automatically distinguish AI contexts from human-written ones.

Recently, with the exponential growth of LLMs, many researchers have attempted to distinguish human-written texts from machine- generated ones. Uchendu et al., 2021, Wang et al., 2024b, He et al., 2024, Liu et al., 2023b have shown us about machine-generated and human writing data from various source. Mitchell et al., 2023, Bao et al., 2023, Deng et al., 2023 used zero-shot classification method to calculate probabilities from perturbated input text. Bhattacharjee and Liu, 2023 leveraged prompt-base method to utilize LLMs as detector. Gehrmann et al., 2019 used statistical method to detect machine-generated paragraph with language model to compute conditional probability. For fine-tuning language model method, Fagni et al., 2021 had a comparative study among pre-trained language model, feature-base and character level classification on DeepFake dataset and showed that pre-trained language model has a best result than others. Liu et al., 2023c used feature-base classification with RoBERTa as embedding and LSTM + Self-Attention in classification head. Bhattacharjee et al., 2023 used unsupervised and self-supervised learning by leveraging domain adaptation on unlabeled dataset and contrastive learning belonging with pre-trained language model to learn domain representations. Kirchenbauer et al., 2023, Liu et al., 2023a used a novel approach with watermark embedding to detect LLMs text that employed by LLMs or neural network.

Being inspired by feature-based classification technique, we propose to have a comparative study for simple and lightweight machine learning method beside the trend of LLM. Our system compares various machine learning models among with ensemble method for multiple machine learning method to find the best combination for our system.

The rest of the paper is as follows. The section 2 generalizes task description and dataset for our experiment. The section 3 shows the description of our system. The experimental setup and results are presented in section 4. Finally, section 5 is the conclusion and discussion about our work.

2 Task description & dataset

2.1 Task description

In SemEval 2024- Task 8 (Wang et al., 2024a), the topic for subtask A is ***Binary Human-Written vs.***

[1] https://openai.com/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 354–360

Machine-Generated Text Classification. The full text is to determine whether an essay is human-written or machine-generated. There are two tracks for subtask A: monolingual (only English sources) and multilingual. On this subtask, we only focus on monolingual dataset.

2.2 Dataset overview

The SubTask-A monolingual dataset originated from various sources of content, including Wikipedia, WikiHow, Reddit, arVix, PeerRead, and OutFox (Koike et al., 2023). According to the author, this is an extended version of the M4 dataset (Wang et al., 2024b). All paragraphs in the dataset of subtask A monolingual are written in English. This dataset contains a total of 159,029 essays, which were split into a three-part train set, development set (dev set), and test set. The monolingual dataset contains two types of labels, 0 represented by human writing and 1 represented by machine-generated. In particular, the train set was constructed from 5 different generator (Human, ChatGPT, Dolly-v2, CoHere and Davinci003) and the development set was constructed only from Bloomz. For the test set, GPT-4 had been added along with the remaining generator to generate essays. The overview statistical and distribution of labels will be detailed in Table 1 and an example of the dataset is represented in Table 2. Additionally, the distribution between human labels and machine-generated labels on the train set is almost equal so the class imbalance technique is not used for this task.

Moreover, a pre-processing step was applied to the dataset by the following deletions and changes:

- Removing punctuation in sentence
- Lower casing text
- Removing any leading and trailing whitespace
- Remove URLs

Dataset	***#Number***	**Label Distribution (%)**	
		Human	Machine
Train set	119,757	52.9	47.1
Dev set	5,000	50.0	50.0
Test set	34,272	47.5	52.5

Table 1: Dataset statistical

3 System Description

We will describe our developed model in this section. On Section 3.1, we will discuss how we embedded sentences in essays using a pre-trained language model. Then, for the crucial section, we would like to present the detail of our model on 3.2. We perform our architecture based on Figure 1. First step, the essays need to be embedded through pre-trained language model. Next, we use ensemble method with base model and then give a final prediction by using meta-model.

3.1 Embedding

For the embedding stage, each token needs to be represented in a vector. Some essays have more than 512 tokens, which will lead to exceed at original BERT (Devlin et al., 2019), we determine to utilize Longformer (Beltagy et al., 2020) model to capture semantic embedding of each word within essays of dataset. Given the essay X, the vector embedding of each token will be calculated in the essay. The input will be formed as (w_i is a word in essay):

$$\texttt{<s>}\ w_1\ w_2\ \ldots\ w_i\ \texttt{</s>}$$

This produces an embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times K}$ (K is a hidden size of word, N is a number of token) by taking the last hidden layer. After that, mean pooling is applied to each vector embedding of the matrix to flatten into a standard vector to aggregate feature of token. The dimension of the vector for each essay will be $\mathbf{X} \in \mathbb{R}^N$ where N is a number of tokens in essay and $\mathbf{X}$ is a feature vector of essay X.

3.2 Ensemble model

After text embedding, we develop our classification stage for the system. We will discuss each base model in section 3.2.1 and how we ensemble various base models in section 3.2.2

3.2.1 Base model

We utilized Support Vector Machine (SVM), XGBoost, Logistic Regression, and K-nearest neighbors (KNN) as the base model for our ensemble method.

Support Vector Machine (SVM) (Hearst et al., 1998) is a supervised learning model that is used for classification and regression tasks. SVM maximizes the hyperplane or set of hyperplanes to find the best boundary that separates different classes in a dataset.

ID	Essays	Label
1	...Step 10. Pause The Game. To pause the game, just press the "start" button...	Machine
56406	...If you haven't used it in the last six months there is little chance you'll use it in the next six months. Toss it.	Human

Table 2: Example dataset

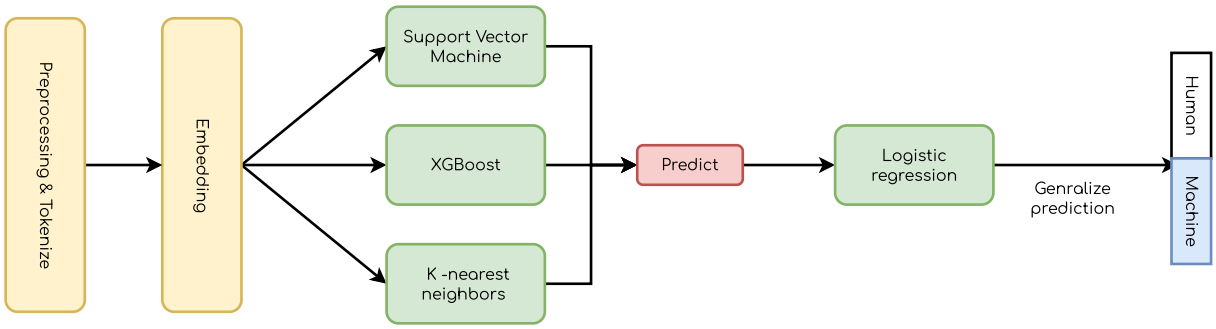


Figure 1: Overview of our system architecture. We demonstrate the best combination model from our experiment

Hyperparameter	Value
C	1.0, 10.0
λ	0.0001, 0.001, 0.1, 0.2, 0.5, 1, 10

Table 3: SVM configuration

Hyperparameter	Values
Learning rate	0.1, 0.2
Estimators	60, 80, 100
Max depth	2, 4, 6

Table 4: XGBoost configuration

Due to the non-linearity of the dataset, we decided to use the Radial basis function (rbf) kernel for SVM, which is defined as Formula 1. Moreover, to find the best parameter for the SVM model, we listed all the hyperparameter values of C and λ used in the grid search as Table 3.

$$K = Ce^{-\gamma||x-z||^2} \quad (1)$$

XGBoost (Chen and Guestrin, 2016) is a scalable end-to-end tree boosting technique which allows to correct the error of the previous tree by creating multiple trees sequentially. The classifier also assigned a weight value to each independent variable and used some techniques to prevent overfit like tree pruning, sparsity awareness, etc.

Same as SVM, we construct candidate hyperparameter values of max depth of the tree and λ used in the grid search as Table 4.

Logistic Regression (LR) is a simple technique for binary classification. Given feature variables, the output is a probability from $[0; 1]$. This can be achieved by applying of a sigmoid function to the linear combination of the independent variables. In our system, we only use the default configuration.

K-nearest neighbour (KNN) is a simple technique for classification which uses majority vote on k closest data point to target point. Grid search is also utilized to find the best value of k, where $k \in \{3, 5, 7, 9\}$.

3.2.2 Stack ensemble

Ensemble different machine learning models is the way to improve prediction accuracy to leverage the strengths and mitigate the weaknesses of the individual base models. In our system, we choose stack ensemble as an ensemble method for our system. For this technique, applicable in scenarios with N base model $(M_1, M_2, ..., M_n)$ and meta-model M, determine meta feature for meta model by predicting each base model $\hat{\mathbf{X}} = (M_1(X), ..., M_n(X))$. Then predict the final output by calculating meta model $y = M(\hat{\mathbf{X}})$. Many different base models and meta model have been evaluated and compared, including Naive Bayes, k-nearest neighbors, SVM, XGBoost, and Logistic Regression as section 3.2.1

3.3 Experiment setup

We describe our system setup procedure. As GPU, we use a single RTX 4090 24GB to train and infer our system for both stages. In the embedding stage, we use Hugging Face[2] library for the Longformer model. For maximum token length in the pre-trained language model, because some essays are longer than 512 tokens, we set it to a maximum of 1024 tokens with padding and truncating. We infer each essay without any training on it.

For the classification stage, with SVM, Logistic Regression, and KNN model, due to the large dimension of the dataset, we proposed to use cuML[3] library, which supports GPU-accelerated for machine learning algorithms. For XGBoost we use xgboost [4] library and sklearn[5] for stack ensemble. Hyperparameter tuning is used in each machine learning model to find the best parameter for each model (all candidate parameters are defined in section 3.2.1). A 5-fold cross-validation is used to find the optimal configuration for the ensemble.

4 Results & Discussion

4.1 Evaluation metric

For subtask A monolingual task, the metrics used to evaluate our result for the dataset are Marco-F1, Mirco-F1, and Accuracy. The main metric for ranking submission is Accuracy. In more detail, the accuracy metric is given by the ratio of the total number of correct predictions to the total predictions done by the model, regardless of true or false predictions. Micro F1-score is the harmonic mean of precision and recall and macro-F1 score is defined as the average of Mirco F1-score across different classes.

4.2 Results

In this section, we present the result of our model, focusing on its accuracy in the dev set and test set. Table 5 shows the performance of our models with some combination with the base model when using the ensemble method mentioned in section 3.2, compared to the dev set. All results were running on the best hyperparameter value of each base model. We first compare the efficiency of each individual model. SVM gives the best performance among all (0.6986). Surprisingly, LR have better performance than XGBoost in monolingual task.

From the result of each model, we also compare our main model with some combinations of the base model when using the stack ensemble method which is represented in Table 5. SVM is used as base models for all ensemble experiments since they give better performance than others. The results do not significantly differ from the three models in our experiment. For model 3, the result is slightly better than model 1 and model 2 which achieved 0.7101 accuracy score.

For the test set, we can notice that all results from 7 methods are not significant differences. The result of SVM (0.8399) still outperformed on individual tests. However, XGBoost has surpassed the performance of KNN and LR on the test set (0.8319 compared to 0.8244 and 0.8155). The LR has the worst performance among 4 models. Surprisingly, after evaluating the test set on the ensemble method, model 1 inferior when compared to the rest. In contrast, model 3 has the best result at the test set with an accuracy of 0.8458. We also visualize our performance of model by representing the confusion matrix in Figure 2

Table 6 shows the result at the stage. We evaluate our result on model 2. Our system achieves 0.8438 which is ranked 36th out of 137. Unfortunately, we can not surpass the result of baseline (achieves 0.8847), which is using RoBERTa model (Zhuang et al., 2021) for classification. Besides, this is a prospective result that can achieve to acceptable score when comparing the traditional machine learning method with the pre-train language model and LLMs. Moreover, we can have an insight into training on traditional machine learning methods and language models nowadays. We believe that if we have a better strategy on hyperparameter tuning, the result could be higher than our official submission.

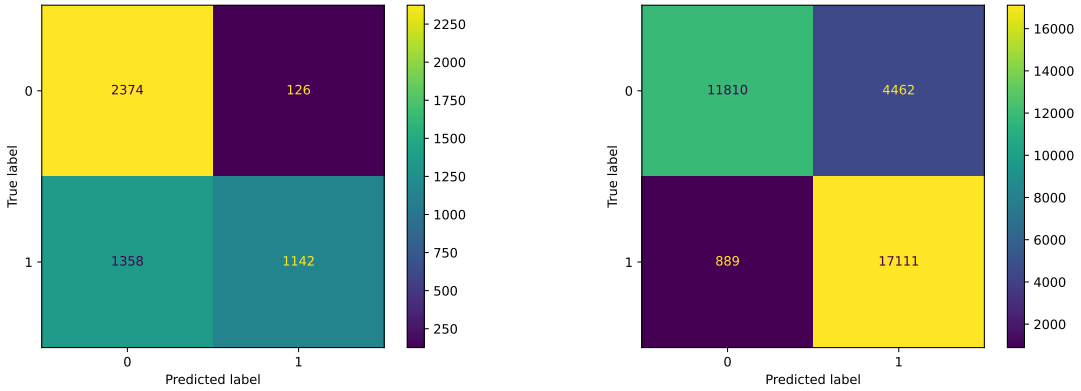
5 Conclusion

In subtask A monolingual of SemEval task 8, we have represented our system for machine-generated detection. We proposed to develop our system based on ensemble of multiple traditional machine learning method with hyperparameter tuning. We found that XGBoost, SVM and KNN as base model and Logistic Regression in meta model would give the highest result. Our official system was ranked the 36th to 137 in test set of subtask A monolingual

[2] https://huggingface.co/
[3] https://github.com/rapidsai/cuml
[4] https://github.com/dmlc/xgboost
[5] https://scikit-learn.org/

Method	Development phase			Test phase		
	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1
SVM	**0.6986**	**0.6986**	**0.6758**	***0.8399***	***0.8399***	***0.8360***
XGBoost	0.6486	0.6486	0.6206	*0.8319*	*0.8319*	*0.8293*
KNN	0.5492	0.5392	0.5057	*0.8244*	*0.8244*	*0.8228*
LR	0.6774	0.6774	0.6549	*0.8155*	*0.8155*	*0.8114*
Model 1	**0.7108**	**0.7108**	**0.6925**	*0.8329*	*0.8329*	*0.8279*
Model 2	0.7032	0.7032	0.6840	0.8439	0.8439	0.8401
Model 3	0.7028	0.7028	0.6832	***0.8458***	***0.8458***	***0.8422***

Table 5: Result on different method on dev set and test set. Test result with italicized have been run after test phase deadline. Denoted that Model 1 is XGBoost + SVM as base model, LR as meta model, Model 2 is XGBoost + SVM + KNN as base model, LR as meta model, Model 3 is LR + SVM + KNN as base model, XGBoost as meta model.



(a) (b)

Figure 2: Confusion matrices for (a) the development set and (b) the test set on official submission

Team	Subtask A - Monolingual Accuracy
Baseline	0.8847
#1 Team	**0.9688**
Ours (36th)	0.8438

Table 6: Result and ranking on test set

with 0.8439 accuracy score and 0.7032 accuracy score in dev set. From our result, traditional machine learning methods still have been proven effective in classification, with some training strategy, compared to other methods such as LLMs.

Acknowledgements

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this research.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Amrita Bhattacharjee and Huan Liu. 2023. Fighting fire with fire: Can chatgpt detect ai-generated text?

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, KDD '16. ACM.

Zhijie Deng, Hongcheng Gao, Yibo Miao, and Hao Zhang. 2023. Efficient detection of llm-generated texts with a bayesian surrogate model.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples.

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2023a. An unforgeable publicly verifiable watermark for large language models.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023b. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023c. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

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.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. 2024a. 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, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji,

Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.