Knowdee at BLP-2023 Task 2: Improving Bangla Sentiment Analysis Using Ensembled Models with Pseudo-Labeling

Xiaoyi Liu, Teng Mao, Shuangtao Yang, Bo Fu
Lenovo Knowdee (Beijing) Intelligent Technology Co., Ltd., Beijing, China
{liuxy, maoteng, yangst, fubo}@knowdee.com

Abstract

This paper outlines our submission to the Sentiment Analysis Shared Task at the Bangla Language Processing (BLP) Workshop at EMNLP 2023 (Hasan et al., 2023a). The objective of this task is to detect sentiment in each text by classifying it as Positive, Negative, or Neutral. This shared task is based on the MUltiplatform BAngla SEntiment (MUBASE) (Hasan et al., 2023b) and SentNob (Islam et al., 2021) dataset, which consists of public comments from various social media platforms. Our proposed method for this task is based on the pretrained Bangla language model BanglaBERT (Bhattacharjee et al., 2022). We trained an ensemble of BanglaBERT on the original dataset and used it to generate pseudo-labels for data augmentation. This expanded dataset was then used to train our final models. During the evaluation phase, 30 teams submitted their systems, and our system achieved the second highest performance with F1 score of 0.7267. The source code of the proposed approach is available at https://github.com/KnowdeeAI/blp_task2_knowdee.git.

1 Introduction

While English dominates as the most resource-rich language in the Natural Language Processing (NLP) community, Bangla which ranked as the 6th most spoken language still faces resource scarcity. Despite three decades of BNLP research, progress has lagged mainly due to scarce resources and associated challenges (Alam et al., 2021).

The objective of the Sentiment Analysis Shared Task is to detect sentiment in each text by classifying it as Positive, Negative, or Neutral. This task utilizes a combined dataset of MUltiplatform BAngla SEntiment (MUBASE) (Hasan et al., 2023b) and SentNob (Islam et al., 2021). MUBASE contains manually annotated social media posts from Twitter and Facebook labeled with sentiment polarity. SentNob consists of social media comments from multiple platforms related to news and videos covering 13 different domains (Islam et al., 2021).

Bangla is a language with rich morphology, many dialects, and unique linguistic nuances. (Alam et al., 2021). Additionally, the dataset used consists of noisy social media comments with a mix of dialects and grammatical errors (Islam et al., 2021). The combination of Bangla's inherent linguistic challenges and the informal, non-standard nature of the dataset creates difficulties for sentiment analysis.

In this work, we present our solution and experimental attempts at the sentiment analysis shared task in Section 2, Our main approach involves an ensembling technique with pseudo-labeling to maximize performance given the limited training data. Results and analysis are followed in Section 3. Finally, Section 4 concludes with a summary of results and an outlook on future directions to advance low-resource natural language processing tasks for Bangla and other languages.

2 System Description

We discuss our proposed solution for the shared task from Section 2.1 to Section n three steps: 1) finetuning an ensemble of models on the provided supervised training data, 2) Using the ensemble models from step 1 to generate pseudo-labels for unlabeled data, 3) Training a new ensemble on the combination of the original training data and pseudo-labeled dataset, to make final predictions.

Additionally, we discuss other pre-trained models we experimented using the proposed solution and another attempted solution in Section 2.4. The experiments result is discussed in Section 3.2.

2.1 Supervised Finetuning

The first step of our solution was to finetune pretrained language models on the downstream sentiment classification task using the provided training data. We split the training data equally into 10 folds.

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 273–278

And we finetuned the same base language model 10 times, using a different fold for validation and the remaining 9 folds for training each time. This generated an ensemble of 10 finetuned classifiers, each trained on a unique subset of the data.

Additionally, we incorporated the Fast Gradient Method (FGM) as an adversarial training technique to improve model robustness and prevent overfitting during finetuning. FGM works by adding small perturbations to the input embeddings based on gradient of the loss. The adversarial noise injections force the model to learn more generalizable representations. The basis of our solution is BanglaBERT (Bhattacharjee et al., 2022), which is a BERT-based language model pre-trained in Bangla using Google Research's ELECTRA (Clark et al., 2020). ELECTRA is a method for efficient self-supervised language representation learning, which can be used to pre-train transformer networks. Specifically, ELECTRA models are trained with the Replaced Token Detection (RTD) objective – to identify which tokens in an input sequence have been replaced by plausible alternatives generated by a small neural network.

2.2 Data Augmentation

After finetuning the 10 models, we utilized them to generate pseudo-labels for unlabeled data as a mean of dataset expansion. The models made predictions on the provided test set, along with confidence scores for each of the 3 sentiment labels per sample.

For each test sample, we summed the confidence scores predicted across the 10 models separately for each sentiment label. If the highest accumulated confidence score exceeded our predefined threshold, we added that sample to the pseudo-labeled dataset with its highest scored label. The higher the threshold is set, the fewer samples are selected for the pseudo-labeled dataset, as only those with very high confidence in the majority of models will pass the cutoff. To obtain a pseudo-labeled dataset with more reliable labels, we set a stringent threshold of 9 out of ten. This ensured that only samples for which the majority of models were highly certain about the sentiment label (the average of the 10 models' confidence scores on the selected label was 0.9 or higher) would make it into the pseudo-labeled set. Samples where the maximum confidence score fell below the threshold were discarded and not added to the pseudo-labeled data.

2.3 Generating final predictions

After creating the pseudo-labeled dataset, we augmented each model's original training set with this pseudo-labeled dataset. Using this expanded dataset, we repeated the finetuning process described in Section 2.1 to train 10 new finetuned models. Each of these 10 models independently predicted sentiment labels for the test set.

To generate the final predictions, we summed the confidence scores per label across the ensemble for each test sample, similar to our pseudo-labeling approach. However, rather than applying a threshold, we directly assigned the label with the maximum summed confidence score as the final prediction.

The ensemble of 10 models helped mitigate noise and overfitting. Combining models exposed to slightly different data distributions reduced individual idiosyncrasies and enabled more robust predictions. The models were less likely to jointly make incorrect high-confidence predictions on ambiguous samples, improving generalization though the training sets were predominantly shared.

2.4 Attempted Models and solutions

Besides BanglaBERT mentioned in Section 2.1, we also experimented other language models with the same training methodology: 1) MuRIL (Khanuja et al., 2021), a BERT model pre-trained on a large corpus of 17 Indian languages; 2) XLM-RoBERTa (Conneau et al., 2019)), a multilingual version of RoBERTa and is pre-trained on data containing 100 languages; 3) mT5 (Xue et al., 2021), a multilingual T5 pre-trained on dataset covering 101 languages.

In addition to utilizing the original dataset, our study incorporated a reformatting approach to conduct in-context learning with the mT5 and BanglaBERT. This method involved a restructuring of the dataset, imbuing each sample with contextual information. For each case, we selected 3 similar samples and their labels from the training set, one for each sentiment label (positive, negative, neutral). The reconfigured dataset was used to finetune mT5 on a text generation task to predict the sentiment label. For BanglaBERT, we finetuned on sequence classification task. It is worth noting that, aside from the variance in the format of training and test data, all other procedural aspects pertaining to the generation of predictions remain consistent with descriptions in Sections 2.1 and 2.3.

3 Experiments and Results

This section presents the official results of our submitted solution for the sentiment analysis shared task. Additionally, we conducted post-evaluation experiments using the gold standard labels to compare the performance of our submitted system against alternative approaches on the test set.

3.1 Experimental Set-up

Our submitted solutions used banglabert_large[1], but we have experimented with various models of different sizes - banglabert[2], muril-large-cased[3], muril-base-cased[4], xlm-roberta-base[5], and mt5-large[6].

We used different hyperparameter configurations for each data format. For the original format, models were trained for 15 epochs with a batch size of 64, max sequence length of 128 tokens, and a learning rate of 2e-05. For the in-context learning format, models were trained also for 15 epochs, but we decreased the batch size to 16, and increased the max sequence length to 384 tokens and the learning rate to 5e-05 in order to accommodate longer contexts.

We also conducted post evaluation experiments on comparing one round, two rounds, and no rounds of pseudo-labeling on different models. All other hyperparameters were held constant across experiments. For both evaluations on dev and test set, we used the official scorer scripts to score the output.

3.2 Results and Analysis

The official results of the top five ranked solutions and baseline solutions for the sentiment analysis shared task are shown in Table 1. Our submitted system achieved an F1-micro score of 0.7267, which ranked 2nd out of 30 participating systems.

Table 2 shows all our experiment results on dev and test set. Our initial experiments (no pseudo-labeling) with various pre-trained language models showed noticeable differences in performance. Across models, we observed up to 3% variance in F1 scores on the dev set. Banglabert achieved the highest dev F1 at 0.7345 (Exp. 4), while multilingual model xlm-roberta-base performed worst at 0.7076 (Exp. 7). However, on the test set muril-large-cased obtained the best F1 of 0.7307. The poorer performance of xlm-roberta-base compared to BanglaBERT and MuRIL models indicates the importance of language-specific pretraining. While xlm-roberta-base was pretrained on multiple languages, BanglaBERT focused specifically on Bangla pretraining and MuRIL on Indian languages including Bangla. The results show that pretraining on closer languages leads to better transferability for Bangla sentiment analysis.

Ranking	Username	F1-Micro
1	MoFa_Aambela	0.7310
2	Our System	0.7267
3	amlan107	0.7179
4	Hari_vm	0.7172
5	PreronaTarannum	0.7164
-	n-gram Baseline	0.5514
25	Baseline (Majority)	0.4977
29	Baseline (Random)	0.3356

Table 1: Official result of the top five ranked solutions and official baseline solutions

To compare training methods, we finetuned mt5-large to generate labels (Exp 8), achieving F1 scores of 0.7095 (dev) and 0.7070 (test). For in-context learning, we constructed similar examples as context to provide more information. With mt5-large (Exp 9), in-context learning improved over direct generation (Exp 8), with F1 of 0.7189 (dev) and 0.7082 (test). However, with banglabert_large (Exp 11), in-context learning decreased performance versus direct classification (Exp 3), scoring 0.7256 (dev) and 0.6675 (test). In summary, providing relevant examples improved the generative task but not the classification task. However, classification still outperformed generation on this shared task.

Based on the above experimental results, we chose classification-based training using banglabert_large for further optimization. we experimented with pseudo-labeling methods. Experiment 1 added 1 round, results show improvement over no pseudo-labeling. Experiment 2 added 2 rounds but gained little versus 1 round, slight dev F1 increase, slight test decrease. Pseudo-labeling boosted performance over no augmen-

[1] https://huggingface.co/csebuetnlp/banglabert_large

[2] https://huggingface.co/csebuetnlp/banglabert

[3] https://huggingface.co/google/muril-large-cased

[4] https://huggingface.co/google/muril-base-cased

[5] https://huggingface.co/xlm-roberta-base

[6] https://huggingface.co/google/mt5-large

ID	Base Model	Training Objective	# of Pseudo-Labeling Rounds	F1-Micro on Dev Set	F1-Micro on Test Set
Original Data Format					
1	banglabert_large	Classification	1	0.7376	0.7267
2	banglabert_large	Classification	2	**0.7384**	0.7224
3	banglabert_large	Classification	0	0.7311	0.7242
4	banglabert	Classification	0	0.7345	0.7236
5	muril-large-cased	Classification	0	0.7303	**0.7307**
6	muril-base-cased	Classification	0	0.7179	0.7081
7	xlm-roberta-base	Classification	0	0.7076	0.7033
8	mt5-large	Generation	0	0.7095	0.7070
In-Context Learning Data Format					
9	mt5-large	Generation	0	0.7189	0.7082
10	banglabert_large	Classification	0	0.7256	0.6675

Table 2: Performance comparison of the submitted solution (shaded) and alternative approaches.

ID	Base Model	Training Objective	# of Pseudo-Labeling Rounds	F1-Micro on Dev Set	F1-Micro on Test Set
1	banglabert_large	Classification	0	0.7311	0.7242
2	banglabert_large	Classification	1	0.7376	**0.7267**
3	banglabert_large	Classification	2	**0.7384**	0.7224
4	xlm-roberta-base	Classification	0	0.7076	0.7093
5	xlm-roberta-base	Classification	1	0.7141	0.7155
6	xlm-roberta-base	Classification	2	**0.7225**	**0.7246**
7	muril-large-cased	Classification	0	0.7303	0.7307
8	muril-large-cased	Classification	1	0.7353	0.7355
9	muril-large-cased	Classification	2	**0.7397**	**0.7401**

Table 3: Pseudo-labeling performance from different models

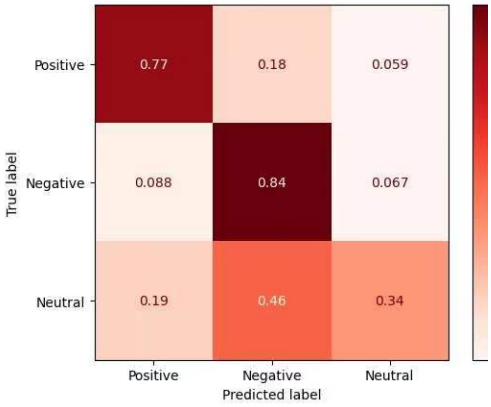

Figure 1: Test set confusion matrix

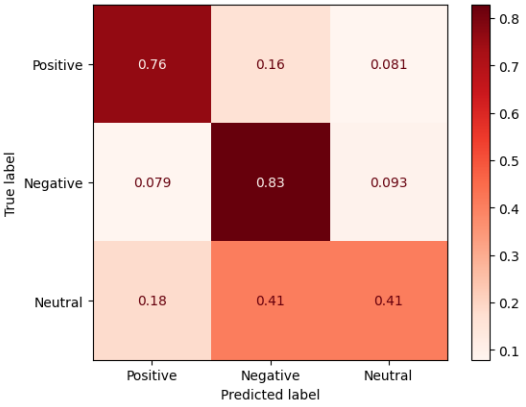

Figure 2: Test set confusion matrix after Pseudo-Labeling

tation. However, increasing from 1 to 2 rounds brought marginal gains on dev, marginal losses on test. This suggests 1 round sufficiently improves banglabert_large on this dataset, while additional rounds may lead to overfitting.

In order to validate the effectiveness of the pseudo-labeling method using ensemble models, we conducted experiments on three models - banglabert_large, xlm-roberta-base and muril-large-cased. The detailed experimental results are shown in the table 3. The results show that for most models, 1 to 2 rounds of pseudo-labeling led to improved performance on both dev and test sets. The banglabert_large model, the model that we submitted to the leaderboard during the evaluation period, achieved the best F1-Micro of 0.7384 on the dev set after 2 rounds of pseudo-labeling. Overall, the experimental results validate that the pseudo-labeling method can effectively improve pretrained language models' performance on downstream tasks.

We also visualized the results on the test set using confusion matrices. Figure 1 shows the confusion matrix for the predictions of the ensemble banglabert_large model on the test set. Figure 2 presents the confusion matrix for the banglabert_large ensemble model after pseudo-label training. Through comparing the two confusion matrices, it can be observed that the model performed relatively poorly on the neutral class - the banglabert_large model achieved an F1 of only 0.34 for the neutral category. After applying the model ensemble pseudo-labeling algorithm, the F1 for the neutral class improved to 0.41. The visualization via confusion matrices and comparison between the banglabert_large model before and after pseudo-labeling provides insight into the performance gain on the challenging neutral sentiment class through utilizing model ensembling and pseudo-labeling.

4 Conclusion and Future Work

In this work, we presented our approach and results for the Sentiment Analysis Shared Task. Our proposed solution using banglabert_large achieved strong performance, ranking 2nd out of 30 submitted systems with an F1-micro score of 0.7267. Through post-competition analysis, we found that larger transformer models designed specifically for Indian languages, such as BanglaBert and Muril, lead to better performance on this multi-class sentiment analysis task.

For low-resource languages like Bangla, pretrained models tailored to the specific language are crucial, as our results demonstrated the superior performance of Bangla-focused models over multilingual models. However, when languages have limited resources, starting with multilingual models from related language families can provide an initial boost, as evidenced by the strong test set results of muril-large-cased pretrained on Indian languages.

As resources grow, continued pre-training of language-specific models on larger and more diverse corpora for that language can further improve adaptation. Additionally, leveraging semi-supervised approaches and generative data augmentation techniques to expand limited labeled datasets will become more viable. Techniques like consistency training, backtranslation, and synthetic data generation can help in low-resource scenarios but require a certain data baseline to be effective.

References

Firoj Alam, Md Arid Hasan, Tanvir Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

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.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

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.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer.