Qualcomm-KAIST Innovation Award 2023

Khuong Le
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon, South Korea
khuonglm@kaist.ac.kr

Abstract

The MBTI classification is a widely recognized task in Machine Learning, serving various practical applications in real-world scenarios. This report introduces a novel solution for this task by leveraging pre-trained BERT-based NLP models for text classification. The proposed approach involves directly classifying text into one of the 16 MBTI classes, followed by ensemble learning, which combines predictions from multiple models to make the final prediction. Despite the limited dataset, the ensemble learning model demonstrates remarkable performance in both Phase 1 and Phase 2 of the hackathon, showcasing its effectiveness in addressing the MBTI classification task.

1. Problem

1.1. Phase 1

In Phase 1, the prediction of the MBTI for each user is necessitated by utilizing a single question-answer pair, along with additional user information such as gender and age. However, this task presents challenges due to the limited scope of a single question-answer pair. Some of these pairs directly inquire about only one specific class within the MBTI, failing to provide insights into the remaining three classes required for accurate MBTI prediction. Consequently, the prediction may lack precision in numerous cases.

1.2. Phase 2

Phase 2 aims to predict the MBTI of each user using a set of question-answer pairs, along with the user's gender and age. This expanded dataset provides more information, allowing the model to learn and predict all four classes in MBTI with higher precision. However, the results of my model in Phase 2 are actually worse than those in Phase 1. This can be attributed to the fact that, although the number of data instances per user has increased, the number of users remains the same in both phases (240 users). In Phase 2, the distinct dataset items are noticeably smaller since multiple question-answer pairs have to be merged into a single data input.

2. Dataset

2.1. Dataset Summary

2.1.1. Question Dataset

The Question data set comprises 60 rows, and each row contains three columns: (1) the index of the data, (2) the index of the question, and (3) the detailed question written in Korean.

2.1.2. Phase 1

The training dataset consists of 11,520 rows of data, where each row contains seven columns. These columns include:

1. Data ID: A unique identifier for each data entry.
2. ID: An integer ranging from 1 to 240, used to identify the user.
3. Gender: A binary value (0/1) indicating the user's gender.
4. User's Age: Categorical values representing the user's age (20/30/40).
5. MBTI Personality Type: The MBTI personality type (e.g., INFP, ESTJ) of the user.
6. Question Number: The index corresponding to the question in the Question dataset.
7. Answer: The user's response to the given question, consisting of a short answer (<그렇다> for 'Yes', <중립> for 'I do not know', <아니다> for 'No'), followed by a descriptive answer written in Korean.

Each user is required to provide answers for questions 1 to 48, and the dataset encompasses information for all users and their respective responses.

2.1.3. Phase 2

The training dataset consists of 7,200 rows of data, where each row contains eight columns. These columns include:

1. Data ID: A unique identifier for each data entry.
2. ID: An integer ranging from 1 to 240, used to identify

the user.

3. Gender: Categorical values indicating the user's gender (male/female).

4. User's Age: Categorical values representing the user's age (20/30/40).

5. MBTI Personality Type: The MBTI personality type (e.g., INFP, ESTJ) of the user.

6. Question Number: The index corresponding to the question in the Question dataset.

7. Short Answer: The user's short response to the given question, where <그렇다> represents 'Yes', <중립> represents 'I do not know', and <아니다> represents 'No'.

8. Long Answer: A detailed and descriptive answer provided by the user for the question corresponding to the index in the Question dataset.

Each user is required to provide answers for questions 1 to 60, and the dataset includes information for 120 random users along with their respective responses.

2.2. Data Preparation

2.2.1. Understanding Phase 1 Dataset

There are four primary classes in the MBTI: I/E (Introvert/Extrovert), S/N (Sensing/Intuition), T/F (Thinking/Feeling), and J/P (Judging/Perceiving). These four classes are further divided into a total of 16 specific classes, as illustrated in Figure 1. Additionally, the dataset distribution of these 16 MBTI classes is uniformly distributed, mirroring the distribution observed in the general population. Consequently, there is no need to adjust or clean the proportional representation of each MBTI class within the dataset, as it already aligns with the expected uniform distribution.

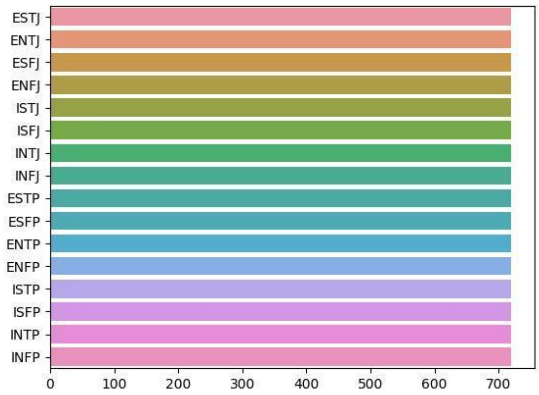

Figure 1. The distribution of each MBTI type

2.2.2. Understanding Phase 2 Dataset

The MBTI consists of four primary classes: I/E (Introvert/Extrovert), S/N (Sensing/Intuition), T/F (Thinking/Feeling), and J/P (Judging/Perceiving). These four classes further branch out into a total of 16 distinct classes, as depicted in Figure 2. The dataset distribution of these 16 MBTI classes is approximately uniformly distributed, similar to the distribution observed in the general population, as shown in Figure 2. Given this approximate uniform distribution, there is no need to perform any proportional representation cleaning or adjustment for each MBTI class, as it already reflects the expected distribution.

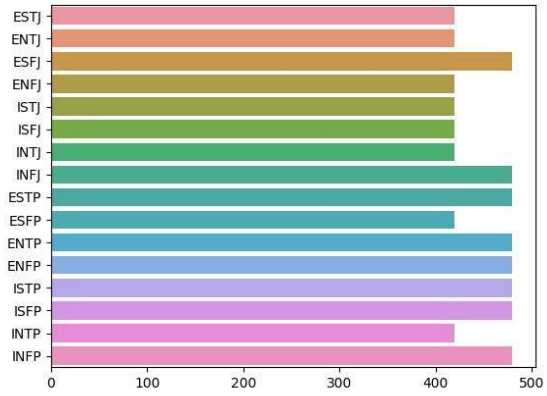

Figure 2. The distribution of each MBTI type

2.2.3. Data Cleaning

Data cleaning plays a crucial role in enhancing data quality and eliminating noise. In line with this, efforts were made to ensure that each word in the data carries maximum meaning. This involved the removal of "stopwords" such as common fillers ('네', '누구', '또', '가') from the list of Korean stopwords [1]. Additionally, special characters and punctuation marks ('<', '>', '.', ',', '/') were also eliminated from the data, further refining its content.

2.2.4. Preprocessing Data

In the data processing phase, several steps were taken to prepare the data for training and validation. Initially, a portion of the data (around 5% to 10%) was set aside as the validation dataset, while the remaining data served as the training dataset for both Phase 1 and Phase 2.

Two different approaches were explored. The first approach involved merging all available information, including the question, answer, age, and gender, into a single dataset for training. This approach yielded decent results initially, prompting the development of the second approach. The motivation behind the second approach was to minimize the potential noise introduced by gender and age. Thus, the training data for this approach consisted solely of questions and their corresponding answers, with gender and age information removed. However, this approach yielded worse results compared to the first approach.

The first approach was deemed better than the second approach for several reasons. Since the training and test data comprised the same users, the gender and age information could be used to match users in the training set with users in the test set. Additionally, considering that MBTI can change over time and that the distribution of MBTI types may vary across different age groups, including gender and age information was deemed beneficial.

In Phase 2, an alternative modification was made to the first approach. Instead of using one question with its corresponding answer separately, multiple continuous questions and answers from the same user were combined. However, due to limitations in token length, only two rows could be merged into one. Attempting to merge three rows resulted in a token length exceeding the limit set by the Kcbert-Large model.

An attempt was made to combine the data from Phase 1 and Phase 2 for training to achieve better results. However, this approach yielded worse outcomes, possibly due to incorrect assumptions made regarding the gender values (0 representing female and 1 representing male) or inconsistencies between the data in Phase 1 and Phase 2.

3. Models

3.1. Model architectures and techniques

3.1.1. BERT

BERT [2] is a pre-trained language model that has had a significant impact on natural language processing tasks. Its innovative bidirectional transformer architecture allows it to effectively capture contextual information from both preceding and following text, leading to a deeper understanding of word semantics. BERT has demonstrated outstanding performance in numerous language understanding tasks and has become a fundamental component in various NLP applications. For this research, the Kcbert-Large [3] model checkpoint was employed as one of the base models, benefiting from its pre-training with online news comments dataset and fine-tuning processes to enhance the study's outcomes.

3.1.2. RoBERTa

In addition to BERT, another powerful model employed in this study was RoBERTa [4], a self-supervised transformers model. RoBERTa incorporates dynamic masking during training, enabling it to effectively utilize both left-to-right and right-to-left contexts. This approach enhances its contextual understanding and language representation capabilities, resulting in superior performance. Empirical findings from the experiment indicate that RoBERTa outperformed other models such as BART, GPT-2, and T5.

In this study, KLUE-RoBERTa-Large [5], a pre-trained RoBERTa model specifically designed for the Korean language, was utilized. KLUE-RoBERTa-Large combines the strengths of RoBERTa with a focus on the intricacies of the Korean language. By leveraging this model, enhanced language understanding and representation were achieved for Korean NLP tasks. The utilization of KLUE-RoBERTa-Large facilitated improved performance and accuracy in the experiments, offering tailored support for the unique characteristics of the Korean language.

3.1.3. ELECTRA

ELECTRA is an innovative pretraining approach that involves training two transformer models: the generator and the discriminator. The generator is responsible for replacing tokens within a sequence and is trained as a masked language model. On the other hand, the discriminator, which is the focus of this study, aims to identify the tokens that were replaced by the generator in the sequence.

ELECTRA stands out from models like BERT or RoBERTa in terms of its training process [6]. While BERT and RoBERTa rely on masked language tasks where tokens are masked and predicted, ELECTRA takes a different approach. Instead of masking tokens, ELECTRA replaces certain tokens with plausible alternatives generated by a compact generator network.

For this study, KoELECTRA v3 [7] was utilized as one of the base models. KoELECTRA v3 is trained on a large corpus of 34GB of Korean text using the WordPiece tokenizer. It provides a powerful foundation for the research, specifically tailored to the Korean language.

3.1.4. Ensemble learning

Ensemble learning is a powerful technique that combines multiple individual models to achieve enhanced generalization performance [8]. Deep learning architectures, known for their superior performance compared to shallow or conventional models, can benefit from ensemble learning. Deep ensemble learning models leverage the strengths of both deep learning and ensemble learning, resulting in a final model that demonstrates improved generalization performance.

In this study, I employed ensemble learning by training three distinct models: Kcbert-Large, KLUE-RoBERTa-Large, and KoELECTRA v3. These models were trained with different settings, including hyperparameter optimizers and learning rates. The outputs of these models were then combined using weighted averaging, where the weights were determined based on their respective performances. This approach allowed me to leverage the strengths of each model and obtain a final result that benefited from the collective predictions of the ensemble.

3.2. Experimental Setup

To ensure effective text processing in this experiment, I utilized BERT-Kor-Base [9] as the word tokenizer. This tokenizer has been specifically trained on a large corpus of Korean text, enabling it to comprehend the intricacies and complexities of the Korean language. The BERT tokenizer operates by breaking down input text into subword tokens, taking into consideration the context and meaning of the words. It employs WordPiece tokenization, which involves splitting words into subword units based on the most frequently occurring subwords in the training data. This approach allows BERT to handle out-of-vocabulary words by decomposing them into subword units that it has already learned during training. By utilizing the BERT tokenizer, I processed the Korean text, breaking it down into a sequence of subword tokens. Each token represents a word or a subword unit that carries contextual information. This enables the preservation of semantic meaning and context, even for complex sentences or phrases.

For training the models, I utilized Kcbert-Large, KLUE-RoBERTa-Large, and KoELECTRA v3. These models were trained with a learning rate of 1e-5 over the course of 10 epochs. The Adam optimizer [10] was employed with a learning rate of 1e-4, and the sparse categorical cross-entropy loss function was used. To assess the performance of the validation dataset, accuracy was employed as the evaluation metric.

4. Training Scheme

4.1. Phase 1

4.1.1. Baseline model

Initially, the baseline model is utilized for training, which followed a sequence of steps: feature extraction, training, and prediction. Each data input was processed by extracting the question, short answer, long answer, age, and gender. These components were combined into a single input string, with the separator '[SEP]' used to distinguish each part. As the main model utilized in the baseline was BERT-based, this input string was then fed into the model for further processing.

The feature extraction step employed a pre-trained BERT checkpoint called kykim/bert-kor-base. This allowed the model to extract relevant features and generate output tensor matrices. Subsequently, these extracted features were trained using a dense neural network. The classifier model employed in this process can be outlined as follows:

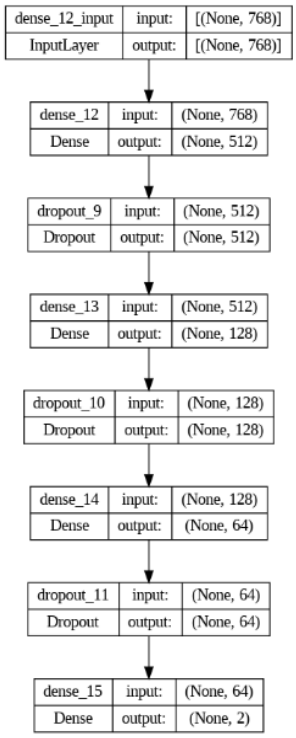

Figure 3. Classifier head

training result for the first MBTI class "I/E"

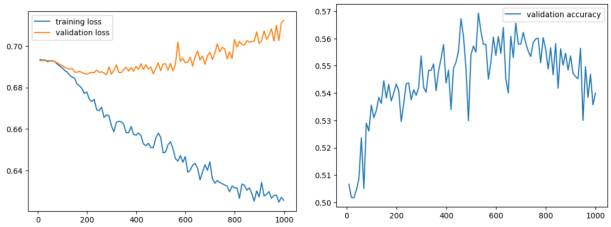

Figure 4. Training & validation loss (left), and validation accuracy (right)

Upon combining the results from the other three classes and conducting the test, I obtained a test result of approximately 0.51, which is comparable to the baseline result. However, upon reviewing the training logs, it became evident that the model struggled to effectively learn from the dataset and exhibited rapid divergence. This suggests that the input data may not have provided sufficient information for effective learning, or there could be issues with the training configuration.

In an attempt to address these challenges, I made several modifications to the training configuration. I increased the dropout probability from 0.1 to 0.5 and then to 0.7, experimented with adding or removing layers in the classifier, and also explored using different BERT-based models such as Bert-base-uncased and distilled-bert-uncased. Unfortunately, these modifications did not lead to improved results. The main limitation was that the head classifier was unable to leverage the embedded model effectively, hindering its ability to derive

deeper patterns from the data.

This suggests that further investigation and experimentation are necessary to identify alternative approaches or modifications that can overcome these limitations and improve the model's performance.

4.1.2. Single Flow Model klue/roberta-large

In this approach, instead of training the model with a separated embed component, a pre-trained BERT-based model, specifically the klue/roberta-large checkpoint, is chosen to train the dataset directly. Due to the large size of the model, training each class in MBTI takes approximately 4-5 hours. Instead of training for each class, a classifier head is implemented to output directly one of the 16 possible combinations of the four MBTI classes.

The training configuration includes a weight decay of 0.01, which matches the value used in the model. The learning rate is set to 1e-5, and a batch size of 8 is used. The dataset consists of both the original data and translated data (Korean -> English -> Korean), and the preprocessing steps are the same as the baseline model. The learning rate scheduler follows a linear schedule.

This approach leverages the power of the pre-trained klue/roberta-large model, which has been trained on a large corpus and can effectively capture the nuances and complexities of the Korean language. By training the model directly with the chosen BERT-based checkpoint, it is expected to learn and generalize well for the MBTI classification task.

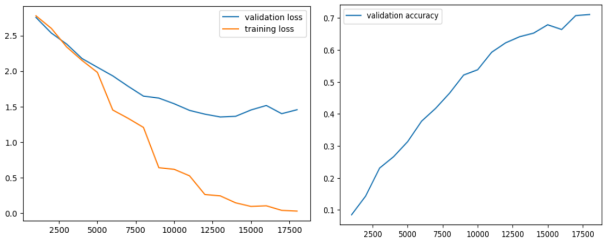

Figure 5. Training & validation loss (left), validation accuracy (right)

The outcome of this approach yielded significant improvement compared to the baseline model, with a test result of 0.71. As part of the experiment, the Single flow mode was retrained using the kykim/bert-kor-base checkpoint. The detailed results can be observed in the provided image:

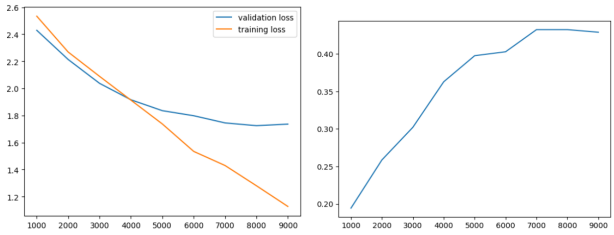

Figure 6. Training & validation loss (left), and validation accuracy (right)

The model's performance is hindered by a tendency to overfit quickly, leading to suboptimal pattern learning compared to the klue/roberta-large checkpoint. Despite efforts to mitigate overfitting by introducing custom dropout probabilities and weight decay, the model's ability to generalize is still limited. This is primarily attributed to the model's smaller size and inferior pre-trained parameters.

To address this issue, the training process continues from the checkpoint obtained at step 19000 of the klue/roberta-large model. The learning rate is reset to 1e-5 and training is continued until the validation accuracy reaches convergence at around 90%. As a result, the test result is improved to 0.73.

During the evaluation of the validation results, it was discovered that there were duplicated data points between the training and validation datasets, which were erroneously split prior to training. This is why klue/roberta model can converge to ~90%. To rectify this, the dataset was rolled back to its original form, and the model is trained again:

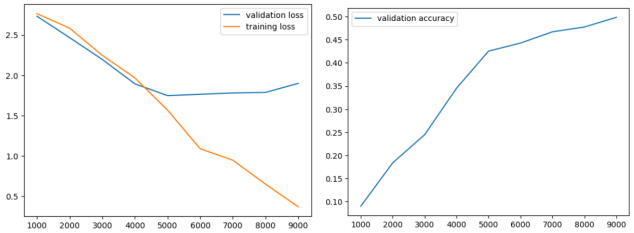

Figure 7. Training & validation loss (left), and validation accuracy (right)

The validation results are greatly improved, test score also increased to 0.8 using prediction at step 6500.

4.1.3. Single Flow Model beomi/kcbert-large

The provided checkpoint is applied to the Single Flow model with the same configuration as mentioned earlier. The training process yielded the following results:

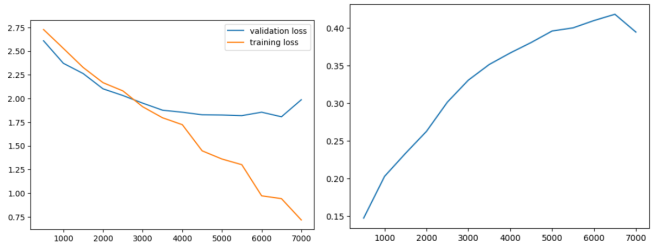

Figure 8. Training & validation loss (left), and validation accuracy (right)

The model was trained on the original dataset without any duplication between the training and validation sets. During training, the validation accuracy reached a convergence point of around 40%, while the test accuracy showed a good result, reaching 0.76.

Various training configurations were tested, including adjusting the learning rate, increasing the batch size, and removing the weight decay. Among these configurations, the removal of weight decay while maintaining the other parameters consistent with the original setup resulted in an enhanced test accuracy of 0.77.

4.1.4. Single Flow Model monologg / koelectra-base-v3-discriminator

The koelectra-base-v3-dicriminator obtained checkpoint is applied to the Single Flow model, using the same configuration as the previous model. Here is the training result:

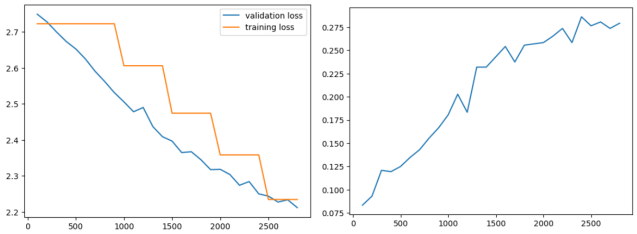

Figure 9. Training & validation loss (left), and validation accuracy (right)

The test result is 0.73.

4.1.3. Ensemble Learning

To obtain the final test result, multiple rounds of training were conducted for each checkpoint. The result files were saved, and the average of these results was calculated. To combine all the models into a single prediction, a weighted average approach was employed. The checkpoints with higher scores on the test dataset were assigned larger weights. For example,

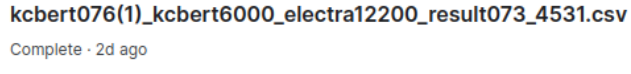

Figure 10. A Phase 1's submission example of ensemble learning

This result is a combination of multiple models, including Kobert, Kcbert (at step #6000), Koelectra (at step #12200), and another Koelectra model. Each model is assigned a weight based on its corresponding test score. In this case, the Kobert model achieved a test score of 0.76, Kcbert achieved a score at step #6000, Koelectra achieved a score at step #12200, and the other Koelectra model achieved a test score of 0.73. The weights assigned to these models are 4, 5, 3, and 1, respectively. By combining the predictions of these models using their weighted average, the overall test result was improved to 0.81. This strategy involved training multiple versions of the three main models and leveraging their collective predictions for enhanced performance on the test dataset.

Applying ensemble learning improved the test score to 0.824.

4.2. Phase 2

In Phase 2 of the hackathon, the three models from Phase 1 are utilized with the same training configurations. The dataset preprocessing remains unchanged, except that two data inputs are combined into a single input during training. A linear learning rate scheduler is employed, and the total training epoch is set to 20.

Similarly, in the test dataset, two consecutive data inputs are merged into a single input item for model prediction. The MBTI label for each user in the test dataset is determined by averaging their question-answer set, which consists of 30 items (as the two items are merged into one). This approach ensures that the MBTI label for each user in the test dataset is representative of their overall responses.

4.2.1 Single Flow Model klue/roberta-large

In Phase 2 of the experiment, the learning rate is set to 1e-6. The training dataset used for this phase is a combination of the dataset from Phase 1 and the dataset from Phase 2. By merging these two datasets, the model can benefit from a larger and more diverse training data, potentially improving its performance and generalization abilities. The combined dataset allows the model to learn from a wider range of examples and patterns, contributing to its overall effectiveness in predicting the MBTI labels.

Test score: 0.52

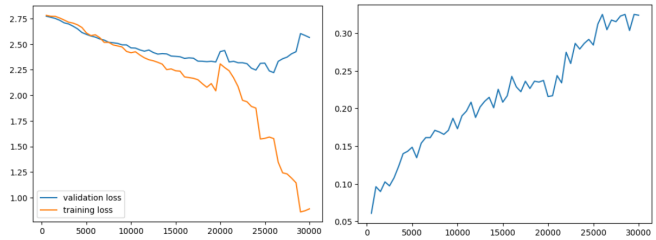

Figure 11. Training & validation loss (left), and validation accuracy (right)

Training the model solely on the Phase 2 dataset while setting the learning rate to 1e-5 leads to an improvement in the validation loss and accuracy, but exectedly has lower test result of 0.49.

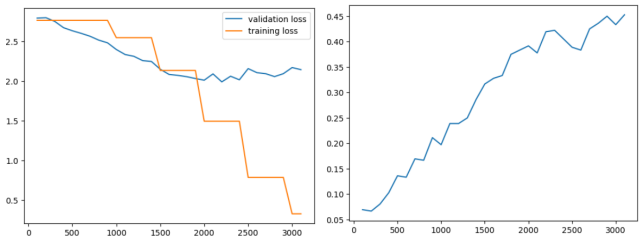

Figure 12. Training & validation loss (left), and validation accuracy (right)

4.2.2. Single Flow Model beomi/kcbert-large

The learning rate used during deployment was set to 1e-5, and only the phase 2 dataset was utilized. All other configurations remained unchanged from the previous model. Training result:

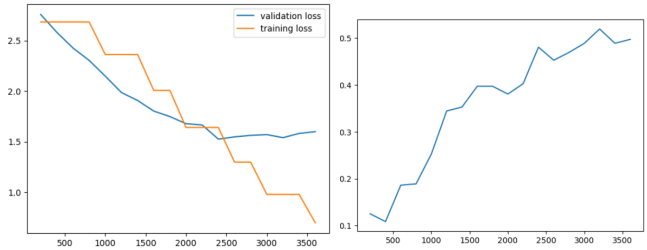

Figure 13. Training & validation loss (left), and validation accuracy (right)

Test score: 0.53

Despite conducting multiple experiments with different hyperparameters such as learning rate, weight decay, batch size, and number of epochs, no significant improvement in results was observed. Subsequently, the model was retrained using a combined dataset comprising both Phase 1 and Phase 2 data, but this did not yield noticeably different training outcomes or test scores.

4.2.3. Ensemble learning

Prediction from the two above models are combined by taking a weighted average and yield a slightly better test score of 0.533

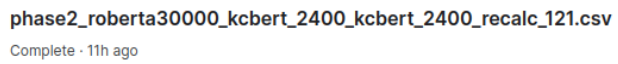

Figure 14. A Phase 2's submission example of ensemble learning

5. Prediction Results

All the models utilized during training were based on a Single Flow structure, where each model directly predicted one of the 16 possible MBTI (Myers-Briggs Type Indicator) outcomes. In an attempt to improve performance, certain models were re-implemented to focus on training exclusively for a single MBTI class. However, this modification did not yield satisfactory results compared to the original Single Flow structure, despite consuming more time and resources for training. On the other hand, in both phases of the training process, combining predictions from multiple models through ensembling proved to be highly effective in achieving superior results on the test dataset.

5.1. Phase 1

Model	Test Acc
Roberta-large	0.80
Kcbert-large	0.771
KoElectra-discriminator	0.73
Ensemble	**0.824**

5.1. Phase 2

Model	Test Acc
Roberta-large	0.51
Kcbert-large	0.530
Ensemble	**0.533**

6. Approach Innovations

Throughout my research, I have explored various approaches that have yielded promising results, and I plan to further enhance these ideas in my future work. One of the techniques I employed was back translation, which involves augmenting the data by generating synthetic examples through translation. This approach has shown the potential in improving the performance of my models by diversifying the training set. Additionally, I successfully applied ensemble learning to achieve higher scores on the test set. By combining the predictions of multiple models, I leveraged their diverse approaches to obtain more robust and accurate results, mitigating the risk of overfitting.

Furthermore, I incorporated language models specifically tailored for the Korean language to enhance the quality of my models. These language models, trained on Korean text, allowed me to capture the unique linguistic nuances and patterns present in Korean. This led to improved performance and accuracy by addressing challenges such as

word order, syntax, and idiomatic expressions that are specific to Korean. By leveraging language models designed for the target language, I was able to fine-tune my models and improve their proficiency in understanding and generating Korean text.

These approaches, including back translation, ensemble learning, and utilizing language models specific to the Korean language, have demonstrated their potential and will serve as valuable foundations for my future work.

7. Failure

In my research, I investigated both data-centric and model-centric approaches. Initially, I pursued a data-centric approach by translating the Korean data and utilizing English language models to tackle the problem. Unfortunately, this approach did not yield satisfactory results, as the performance of the models was below expectations. To overcome this limitation, I implemented the back-translation technique, involving the translation of Korean data to English and then back to Korean. However, empirical results indicated that although this technique increased training time, it did not significantly improve performance. Moreover, it introduced more noise and bias, particularly in tasks involving multi-label classification [11].

In my exploration of the model-centric approach, I experimented with various language models such as T5, BART, and GPT-2. However, none of these models delivered desirable results. As an alternative strategy, I decided to employ four separate single binary-class models. The objective was to capitalize on the strengths of each model and combine their predictions to achieve a more accurate overall outcome. The decision to utilize four binary-class models was motivated by the aim to address potential imbalances or intricacies within the dataset that could be better handled by specialized models. By focusing on binary classification for each class, I aimed to capture specific patterns and nuances unique to each category. However, after extensive experimentation and evaluation, it became evident that this approach resulted in suboptimal performance compared to training a single 16-class model.

During my exploration of ensemble learning techniques, I initially assigned equal weight to each model by averaging their results. However, after evaluating the validation accuracy, it became clear that certain models outperformed others. To account for these disparities, I decided to increment the weight of the superior models rather than uniformly assigning the same weight to all models.

8. Conclusion and Future Work

This report highlights the effective application of pre-trained BERT-based models in predicting MBTI types based on question-answer surveys. It is observed that larger-sized models have a greater capacity to extract patterns from the data, leading to improved generalization. Additionally, ensembling the predictions from multiple models significantly enhances the performance of the test dataset.

However, it is noted that the results of Phase 2 did not surpass or match the performance of Phase 1, contrary to expectations. This discrepancy could potentially be attributed to the small size of both the training and testing datasets, consisting of only 120 different users each. Such limited data size may introduce fluctuations and inconsistencies in the results.

Future work will involve conducting experiments with alternative pre-trained NLP checkpoints, exploring different ensembling methods, and working with larger datasets. These endeavors aim to further enhance the predictive accuracy and robustness of the models used in this study.

References

[1] Spikeekips, Korean Stopwords, 2016. Retrieved from https://gist.github.com/spikeekips/40eea22ef4a89f629abd87eed535ac6a

[2] Delvin J., Chang M.W, Lee K., Toutanova K., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019. Retrieved from https://arxiv.org/pdf/1810.04805.pdf

[3] Lee J., 2020. Retrieved from https://huggingface.co/beomi/kcbert-large

[4] Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V., RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. Retrieved from https://arxiv.org/pdf/1907.11692.pdf

[5] Park S., Moon J., Kim S., Cho W.I, KLUE: Korean Language Understanding Evaluation, 2021. Retrieved from https://arxiv.org/pdf/2105.09680.pdf

[6] Cortiz D., Exploring Transformers models for Emotion Recognition: a comparision of BERT, DistilBERT, RoBERTa, XLNET and ELECTRA, 2022. Retrieved from https://dl.acm.org/doi/10.1145/3562007.3562051

[7] Park J., 2020. Retrieved from https://huggingface.co/monologg/koelectra-base-v3-discriminator

[8] Ganaie M.A., Hu M., Malik A.K., Tanveer M., Suganthan P.N., Ensemble deep learning: A review, 2022. Retrieved from https://arxiv.org/pdf/2104.02395.pdf

[9] Kim K., 2020. Retrieved from https://huggingface.co/kykim/bert-kor-base

[10] Kingma D.P., Ba J.L., ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION, 2015. Retrieved from https://arxiv.org/pdf/1412.6980.pdf

[11] Edunov S., Ott M., Auli M., Grangier D., Understanding Back-Translation at Scale, 2018. Retrieved from https://aclanthology.org/D18-1045.pdf