QIA Hackathon 2023 report

Diyar Tulenov
KAIST
diar5tulenov@kaist.ac.kr

Zhantai Dzhusupov
KAIST
jantaidjusupov@kaist.ac.kr

Abstract

We present a unique approach towards predicting the Myers-Briggs Type Indicator (MBTI) based on open-ended questionnaire responses. By utilizing a pre-trained BERT model on a Korean dataset, we introduced two innovative techniques that significantly enhanced our MBTI prediction accuracy. Notably, we found that the prediction accuracy improved when the model was tasked to predict both the user's MBTI type and their specific User ID, despite the fact that we were not directly using the User ID prediction information. We propose a novel method to derive MBTI from User ID predictions, leading to promising results that warrant further exploration.

1. Introduction

The accurate prediction of an individual's Myers-Briggs Type Indicator (MBTI) has proven to be a challenging task due to the inherent complexity of human personality traits. This paper presents our approach used in a recent Kaggle hackathon, where we had to predict MBTI based on questions and answers. To tackle this challenge, we employed a pre-trained BERT model on a Korean dataset, with the addition of unique user-specific prompting and dual-output classification methods.

2. Phase 1

2.1. Methodology

Our approach revolved around two distinct BERT-based models:
Model A, which predicted MBTI using a four-way binary classification output.
Model B, which expanded upon Model A by also predicting the user's ID using a 240-class classification output.
Model C, which only used the user's ID using a 240-class classification output.

For each dataset entry, we generated prompts that encompassed the gender, age of the individual, the posed questions, and the corresponding answers, by simply concatenating all these variables into one string in Korean.

2.2. Model Training and Validation

We partitioned our dataset for model training and validation purposes. The training set included the first 40 questions, while the validation set comprised questions 41 to 48. We subsequently trained the BERT transformer with our classification heads on top. The optimal tuning parameters and number of training epochs were determined through this iterative process. Post this, the training and validation datasets were consolidated for a final round of model training, and we subsequently generated the output submission file.

2.3. Results

In the assessment of our models, Model A achieved a respectable Area Under the Receiver Operating Characteristic Curve (AUROC) of 0.7. However, when we introduced an additional prediction task - User ID - into Model B, we observed a significant enhancement in AUROC, attaining a value of 0.76. This marked improvement emphasizes the efficacy of our multi-task learning approach, reaffirming the contribution of secondary tasks to the primary objective, in this case, MBTI prediction, even if their link isn't overtly direct.

Further, when the test AUROC was evaluated on Kaggle for the enhanced Model B, it remarkably climbed to 0.78. This performance boost is attributable to a novel derivation technique from User ID predictions. The method entailed the calculation of probabilities for each user, followed by multiplying these with their respective MBTI values. The final prediction for a given input was obtained by summing these weighted MBTI values over

all users, thereby allowing for a nuanced reflection of user-specific tendencies in the MBTI prediction. This approach further substantiates the beneficial role of integrating user-specific information in the prediction of intricate traits such as MBTI.
Taking this user-centric approach a step further, Model C, which was solely dedicated to predicting user IDs, surprisingly demonstrated a marginal yet significant improvement over Model B. This intriguing observation suggests that the primary learning exhibited by our phase 1 models was not necessarily tied to MBTI prediction but was rather centered on discerning individual-specific traits. Hence, the results underscore the potential of user-specific modeling in the accurate prediction of complex traits such as MBTI. Also, the validation accuracy for user_ID's was 30%, which is a high value given that there are 240 users, and the model predictions are based only on one question answer pair.

Last few present improvements in kaggle test AUROC were found using the following method:

1. Once the optimal number of epochs was determined to prevent overfitting, using the initial train-validation split, we proceeded to train our model on the comprehensive dataset, merging both the training and validation subsets. This proved to consistently improve the test score in Kaggle.
2. Using an ensemble of model B MBTI predictions and model C MBTI predictions by using user probabilities. In our case, the mean of predictions was taken.
3. Using a bigger pre-trained model. The Bert Korean model we used has a base and large version. To save computation time and expenses, our initial experiments and fine tuning would be done on a base model, after which we would use the same training pipeline on the large model, generally improving the score by a slight margin.

These small optimization techniques allowed us to marginally improve our method to reach a final AUROC value of 0.81859 we have at the leaderboard at the time.

2.4. Discussion

The main finding of our research lies in the enhanced prediction accuracy when including user-specific information in the test dataset. Essentially, our model was largely predicting the user ID rather than predicting the MBTI directly, leading us to suggest that MBTI prediction could be implicitly associated with user-specific features. However, direct prediction of MBTI without prior knowledge of the user was found to be substantially more complex, as we will explore in Phase 2.

3. Phase 2

3.1. Methodology

The study was initiated with Natural Language Processing (NLP) models trained on the Phase 1 data, where MBTI predictions were based on users' responses to a single question. The models achieved an AUROC score of approximately 0.7.

Subsequently, the same models were applied to the Phase 2 dataset. In this phase, the models were trained with known users' answers, with mean predictions being taken for each question to predict a user's MBTI.

Given the significant decline in performance on the Phase 2 data, NLP methods were temporarily abandoned, and other machine learning models were evaluated, including CatBoostClassifier, DecisionTree Classifier, LightGBM, and Logistic Regression. In addition, an ML-independent approach was tested, which involved using ChatGPT to classify each question for each letter prediction.

3.2. Model Training and Validation

For the CatBoostClassifier, the data was cleaned, and long answers were excluded. Four separate models were created, one for each MBTI dichotomy (I/E, S/N, T/F, J/P). The best validation results were derived from a 20% split of the training data (around 0.6 on validation). However, these results were not as impressive in the final submission

The Logistic Regression model was based on an assumption that the ground truth labels for MBTI originated from a scoring method akin to a 16 personality test. Over 1000 repetitions, train and test data was split randomly.

3.3. Results

In Phase 2, the NLP model exhibited a disappointing AUROC score of approximately 0.5, with no further improvements beyond 0.52. The CatBoostClassifier demonstrated a slightly better performance with an AUROC of 0.528. In contrast, DecisionTree Classifier and LightGBM models yielded similar or inferior results. All these are the submission results.

Interestingly, the Logistic Regression model performed poorly on the S/N, T/F, and J/P categories, despite a high mean AUROC of 0.95 on the training data. The model continually predicted with an accuracy of around 0.39, significantly lower than the expected 0.5.

The ML-independent approach, involving ChatGPT, performed poorly on the Phase 2 dataset but demonstrated a noticeable improvement on the Phase 1 dataset (AUROC is around 0.9), using only 48 questions.

3.4. Discussion

The marked discrepancy in performance between Phase 1 and Phase 2 suggests that these datasets may be distinct in their characteristics. Phase 1 appeared to adhere to a more logical distribution, while Phase 2 seemed more random.

Several factors could account for these differences. For instance, the data in the two phases could have been gathered under differing conditions or with different methodologies. Alternatively, there could be inconsistencies or errors in the labeling of the Phase 2 data.

4. Conclusion

The use of user-specific prompting and dual-output classification significantly improved our MBTI prediction performance for Phase 1. The discovery that the inclusion of user prediction tasks elevates the MBTI prediction performance, despite not directly utilizing the user prediction information, is a novel insight that could revolutionize personality type prediction using machine learning models. Further research is necessary to validate and optimize this approach for broader application for the first Phase. Despite the success of predictive models in Phase 1, their performance significantly degraded in Phase 2. Further investigation is required to understand the nature of the differences between these phases and adjust the model training approach accordingly. Additionally, it's crucial to evaluate the quality and source of the data before developing predictive models.

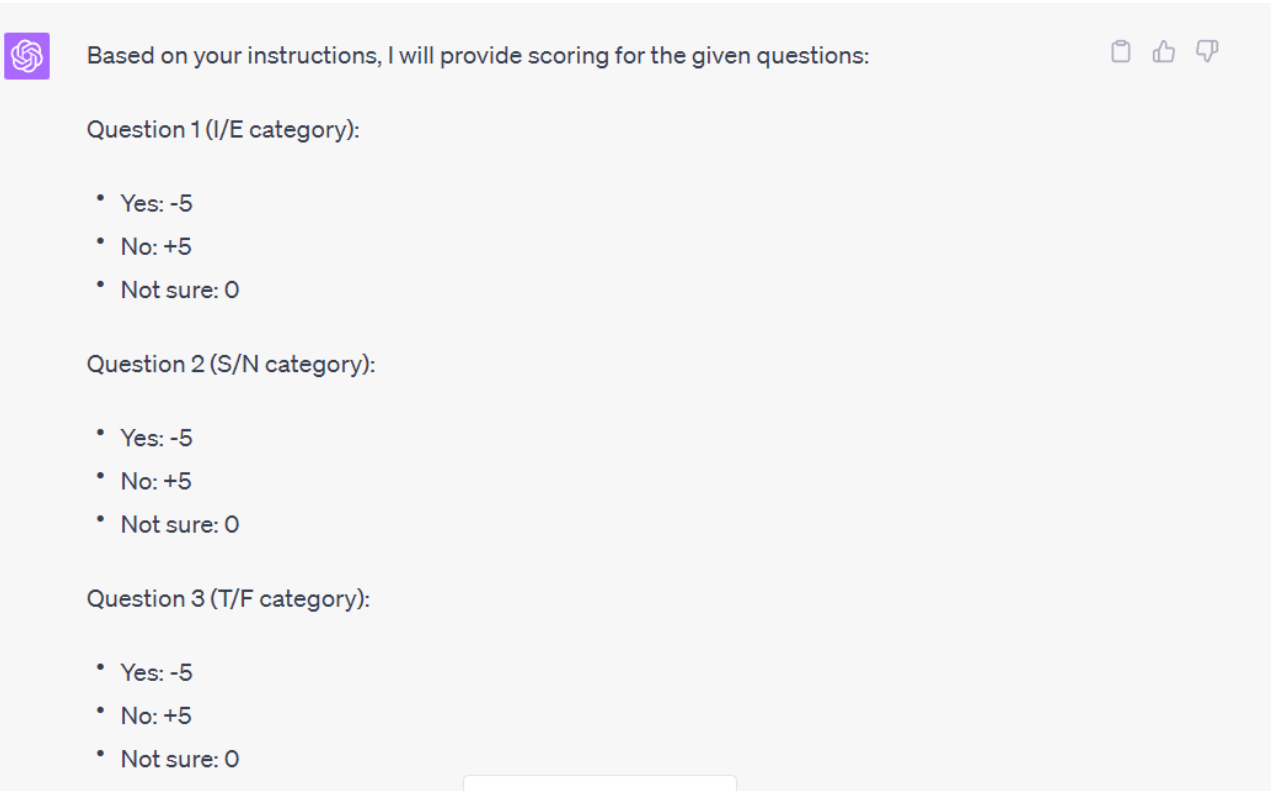
EXHIBIT 1 - Logistic Regression for S/N:

In [348]:
```
ROC=[]
TROC=[]
for i in range(1000):
    X_train, X_test, y_train, y_test = train_test_split(train, Y_train2, test_size=0.2)

# Initialize the logistic regression model
    logreg = LogisticRegression(penalty='l1', C=3, solver='liblinear')

# Train the logistic regression model
    logreg.fit(X_train, y_train)

# Make predictions on the test set
#
#sub2['I/E']=logreg.predict_proba(val)[:, 1]

# Evaluate the model's performance
    y_pred_proba = logreg.predict_proba(X_test)[:, 1]  # Predict probabilities for the positive class
    yy=logreg.predict_proba(X_train)[:, 1]
# Calculate AUROC
    auroc = roc_auc_score(y_test, y_pred_proba)
    #print("AUROC:", auroc)
    ROC.append(auroc)
    auroc = roc_auc_score(y_train, yy)
    TROC.append(auroc)
```

In [349]:
```
np.array(ROC).mean()
```

Out[349]: 0.3904295582339845

EXHIBIT 2 - Interaction with ChatGPT



Based on your instructions, I will provide scoring for the given questions:

Question 1 (I/E category):

- Yes: -5
- No: +5
- Not sure: 0

Question 2 (S/N category):

- Yes: -5
- No: +5
- Not sure: 0

Question 3 (T/F category):

- Yes: -5
- No: +5
- Not sure: 0

EXHIBIT 3 - No ML results on Phase 2 dataset

In [470]:

```
from sklearn.metrics import roc_auc_score

auroc1 = roc_auc_score(Y_train1, df_ie['I/E'])
auroc2 = roc_auc_score(Y_train2, df_sn['S/N'])
auroc3 = roc_auc_score(Y_train3, df_tf['T/F'])
auroc4 = roc_auc_score(Y_train4, df_jp['J/P'])

print('AUROC I/E: ', auroc1)
print('AUROC S/N: ', auroc2)
print('AUROC T/F: ', auroc3)
print('AUROC J/P: ', auroc4)
```

```
AUROC I/E:  0.5434722222222222
AUROC S/N:  0.4736111111111111
AUROC T/F:  0.5230619616560155
AUROC J/P:  0.5200222469410456
```

EXHIBIT 4 - No ML results on Phase 1 dataset

In [432]:

```
from sklearn.metrics import roc_auc_score

auroc1 = roc_auc_score(Y_train1, df_ie['I/E'])
auroc2 = roc_auc_score(Y_train2, df_sn['S/N'])
auroc3 = roc_auc_score(Y_train3, df_tf['T/F'])
auroc4 = roc_auc_score(Y_train4, df_jp['J/P'])

print('AUROC I/E: ', auroc1)
print('AUROC S/N: ', auroc2)
print('AUROC T/F: ', auroc3)
print('AUROC J/P: ', auroc4)
```

```
AUROC I/E:  0.9185069444444444
AUROC S/N:  0.8773263888888889
AUROC T/F:  0.8512847222222221
AUROC J/P:  0.8925000000000001
```