COMS 5790 : Project 2

Rohail Alam
rohail03@iastate.edu
Kaggle Username: rohailalam

1 Method

1.1 Baseline

To establish a baseline for comparison, multiple traditional machine learning models were trained using the given dataset. The input dataset consisted of a combination of the title and abstract for each research paper. By merging these two fields, the goal is to obtain a more representative and compact field for the category of the paper.

A series of preprocessing steps were performed to clean and normalize the text data before training the baseline models. First, irrelevant columns were dropped from the dataframe. Such columns would be the PMID and the Journal. The reason why Journal is being dropped is because there are instances where the journal names are very similar for two papers, but they have different categories. In efforts to not confuse the classifier in such scenarios, it is in best interest to drop this column. The forthcoming steps included Tokenization, and removal of stop words. Additionally, Word Stemming and Lemmatization were applied, which ensured that variations of the same word were treated as a single item. After carrying out these series of preprocessing steps, Bigrams were generated by combining pairs of consecutive words to capture context-based relationships between adjacent words.

Next, TF-IDF vectors were computed for the entire corpus. The vectors were then used as input features for various classifiers, including XGBoost, K-Nearest Neighbors (KNN), Ensemble methods, and Naive Bayes. The performances of these models can be found in the Result section of this document.

The dataset shows a heavy imbalance in classes - there are a lot more relavant papers than irrelevant ones (approximately 1 irrelevant paper for every 10 relevant papers) .Several techniques were employed to mitigate this issue and improve the model's performance. SMOTE (Synthetic Minority Over-sampling Technique) was applied to artificially increase the number of samples from the minority class, and in some cases class weights were used to prioritize the correct prediction of the minority class. Additionally, hyperparameter tuning was performed to optimize the performance of each classifier by selecting the most effective model configurations.

1.2 BERT

While the traditional machine learning models provided a solid baseline, more advanced deep learning approaches were explored to further improve classification performance. Fine-tuned BERT models demonstrated superior results compared to the baseline models. Two different pretrained transformer-based models were tested on the dataset: the base BERT model and SciBERT.

Before training the BERT models, the dataset underwent additional pre-processing similar in the case of the baseline models to ensure consistency and quality. All text was converted to lowercase to eliminate discrepancies due to case differences. Furthermore, special characters and extra whitespace were removed to clean the textual data. Stop words were also removed. The data was then tokenized using BERT's tokenizer, which mapped words into tokens suitable for input into the pytorch transformer model.

To handle class imbalance in the BERT training phase, a rather uncommon strategy was used. Instead of oversampling the minority class, the majority class was undersampled. This decision was made due to the large size of the dataset due to oversampling, which would have significantly increased the computational requirements and training time for the BERT models. By reducing the number of instances in the dominant class, the model could still learn balanced representations without excessively prolonging the training process. In addition, a weighted loss function in the form of binary cross

entropy was also utilized, which also proved to be effective.

Overall, the fine-tuned BERT models achieved better performance than the traditional machine learning approaches, demonstrating its effectivness granted, at the cost of a rather large training time. The importance of class balancing was clearly highlighted in the prediction metrics of the models.

2 Result

The evaluation metrics of the baseline models are shown below. It is to be noted that the model is trained on a class-balanced dataset (using SMOTE) and hyperparameter tuning (some using *GridSearchCV* and some using the *hyperopt* library). The scores here represent the F1 scores on the training and testing datasets.

Classifier	Training Score	Test Score
XGB	0.920	0.658
Naive Bayes	0.887	0.673
KNN	0.943	0.676
Ensemble	0.924	**0.717**

Table 1: Comparison of Baseline Classifiers

It can be inferred from the table that the Ensemble classifier sets the best baseline accuracy of around 71%. This particular classifier is derived from the 'imbalanced learn' python library, which is an ensemble of AdaBoost learners trained on different balanced bootstrap samples.

The metrics for the fine-tuned BERT models are shown below. Every model used is the uncased version, which means it does not distinguish capitalized words from the ones that are not.

Model	Training Score	Test Score
deBERTa	0.97	**0.862**
Base	0.972	0.833
SciBERT	0.94	0.828
DistilBERT	0.966	0.787

Table 2: Comparison of fine-tuned BERT models

While the Base BERT pretrained weights show the best training performance, the deBERTa model did better against the test dataset. This is probably because the BERT model was slightly overfit to the training data. A weighted binary cross entropy loss function was utilized to handle class imbalance along with a classiication threshold of 40%, which proved to be effective since the testing score greatly improved on doing so (from around 0.65 to 0.86)

3 Discussion

More about the oversampling method SMOTE - this is also a part of the 'imbalanced learn' python library and is seen as a better way of oversampling than compared to random oversampling, which was previously utilized for this project. SMOTE generates synthetic samples that emulate the characteristics of the class it belongs to, which would make the model more robust and not fit to only the data provided in training. This method selects the k nearest neighbors belonging to the same class and then interpolates between the original instance and the neighbors. (Bowyer et al., 2011).

Another thing to mention is the use of sciBERT, which is essentially BERT trained on scientific text. The reason why this was tried is because this specifically focuses on the domain of our problem. sciBERT had promising potential, but was somehow unable to outperform the base BERT model, even though both of them were trained on the same dataset.

An attempt to augment the data by synthetically generating classes belonging to the minority label was made. Simply replicating the data did not prove to be effective, as the model just overfit and did not show robustness against the testing data. Instead, an LLM API was utilized to paraphrase each sample and thus generate a new entry with different wording but still captures the main idea. together.ai provides a free to use LLM API, and was fed the following prompt for every entry that belonged to the minority class:

"I want you to generate five additional samples of the text provided to you. The text is the combination of the title and abstract of a relevant paper. Here, the relevant papers mean that these papers contains important geno-trait results that can be curated in the QTLdb. The samples generated should have the content of the entire text provided to you. I want you to ensure that each sample is as unique to each other as possible while prioritizing the keywords and the main idea of the text. Every sample should be printed in a single line and separated by a ' ' delimiter. The prompt output should not have any newline characters and contain only the line of samples and nothing else : (Entry here)"

A Appendix

The following resources were utilized to aid the coding process of the project:

- sciBERT Github Repository
- Thread on handling class imbalance
- Official BERT documentation
- Implementation of deBERTa Model for Multi-class classification
- Fine-tuning BERT for an unbalanced multi-class classification problem

References

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813.