Text Mapping to ICD-O Codes Final Project Report

Haemin Lee haeminle@usc.edu
Kevin Tsang kktsang@usc.edu
Hardi Rathod hrathod@usc.edu
Matthew Brennan mtbrenna@usc.edu

Github Link: `https://github.com/haemin-lee/ICD-O-Classification`

Youtube Video Link: `https://youtu.be/4sFVNDzIPoc`

1 Introduction

Pathology reports contain diagnostic information about tumors described by pathologists examining tissue specimens. The diagnosis description texts, also known as cancer summary, are dictated pathology evidence about the type of cancer, special features of the cancer and whether the cancer has spread outside the organ where it originated. The pathology reports come in two forms: free text and synoptic report. Free texts are the unstructured texts, often transcribed by recordings or text-to-speech devices, described in the final diagnosis section of pathology reports. Synoptic reports are reports that describe specific data elements in a consistent format in surgical pathology and they are concise, organized and contain all necessary data. International Classification of Diseases for Oncology (ICD-O) are created by the World Health Organization and can be used as the ontology for coding the site (topography) and the histology (morphology) of the neoplasm. The systematic and structured ICD-O codes are generally assigned manually, but with the increasing adoption of Natural Language Processing (NLP), one can write programs to assign ICD-O codes to free texts and synoptic reports.

The project goal is to build an open-source classifier with the appropriate encoding method and perform an analysis on whether intentional inclusion of negation affects the ability of the model classifying the right label. In order to create the best classifier for this task, we focused on experimenting modifying parameters for upsampling, encoding and models. In addition, we are interested in exploring how the model handles negation in the data as well as explored why some models worked better than others.

2 Methods

2.1 Preprocessing

In order to make sure the dataset contained clean and relevant data, we preprocessed the data. We removed all non-alphabetical characters, punctuation, and irrelevant white space, then cast all the text to lowercase. The data consisted of free text descriptions that relied heavily on medical terms. There were 49 potential labels, where each label corresponded to a different ICD-O code.

2.2 Embedding

The synoptic text descriptions were encoded into numerical vectors by using BERT encoders. BERT encoders were selected because of their ability to bidirectionally train unlabelled words. We chose to encode the input data using the untrained open source huggingface BERT transformer.

2.3 Upsampling

There was a great imbalance of ICD-O codes in the received dataset, as there were some codes that were greatly represented in the dataset while other codes were vastly underrepresented. In order to construct a more balanced dataset of ICD-O codes, we upsampled the data.

2.4 Model

We experimented with a wide variety of classifiers in order to effectively classify the free text to the corresponding ICD-O code. The final classifier used was a BLSTM with a linear

layer, ELU layer, linear layer, and a softmax layer.

2.5 Testing

In order to test the effectiveness of the classifier, the testing data was split into a train, dev, and test split. When predictions were made on the testing data, the accuracy was 99 percent, precision was 98 percent, recall was 99 percent and the F-1 score was 99 percent.

3 Experiments

3.1 Embedding

3.1.1 Word2Vec

The data was also embedded both with the pretrained Word2Vec transformer as well as a custom Word2Vec model trained with the dataset. This proved to be ineffective, as when a vanilla neural network was trained and tested with the BERT encodings, the accuracy was not much better than random classification. The embeddings generated with the pretrained Word2Vec model classified at 19 percent accuracy while the embeddings generated with the custom Word2Vec classified at 27 percent accuracy.

3.1.2 BERT

The data were embedded with the open source huggingface BERT transformer. This proved to be effective, as it was able to capture relevant relationships between the free text and the ICD-O codes. When a vanilla neural network was trained and tested with BERT encodings, it reached a maximum accuracy of 44 percent. Even though the accuracy was not high, considering that there are 49 different labels, this showed the effectiveness of the BERT transformer.

3.2 Upsampling

Given the nature of the imbalanced classes in the ICD-O codes, the team experimented with upsampling the instances and setting cut-off points for the frequencies of data with each code. We had set the cut off point for the codes to be 10, which means we must have 10 rows of synoptic text data that are assigned to a specific ICD-O code. This is because we predicted that the model will be very noisy due to the low frequency and invariation of the labels with less than 10 instances. With this cut off point, we were able to achieve 98 percent accuracy in the BLSTM model. Subsequently, the group changed the cutoff frequency to 5 and the accuracy became 97 percent.

3.3 Model

3.3.1 LightGBM

LightGBM is a powerful gradient boosting algorithm which uses tree based algorithms to classify data. One of the ways to visualize our problem could be looking for specific words/embeddings and configuring how the occurrences of certain words affect the classification. After upsampling the size of the data was huge and using LightGBM gave us a much better initial result. To implement LightGBM, the data was fed in a parallel fashion and a leaf-wise tree growing mechanism was used. To increase the time efficiency, LightGBMXT was also utilised. The final accuracy after optimization was 98 percent with precision averaging to 0.99 percent.

3.3.2 CatBoost

This is yet another high performing machine learning algorithm to achieve high performance without extensive data training. It works great for tabular predictions and can help achieve great results even in case of highly skewed datasets such as this one.GridSearch was additionally applied in order to choose the best possible combination of the hyper parameters.The final accuracy after Grid Search optimization for hyper parameter was 96 percent with precision averaging to 98 percent.

3.3.3 XGBoost

Similar to above models, XGBoost Model uses a gradient boosting framework for a decision tree based ensemble learning algorithm. It is specifically designed to be faster, prune inactive branches so that we can get efficient models.

3.3.4 MutiStacked NN model with multiple BERT Embeddings and FastText Embedding

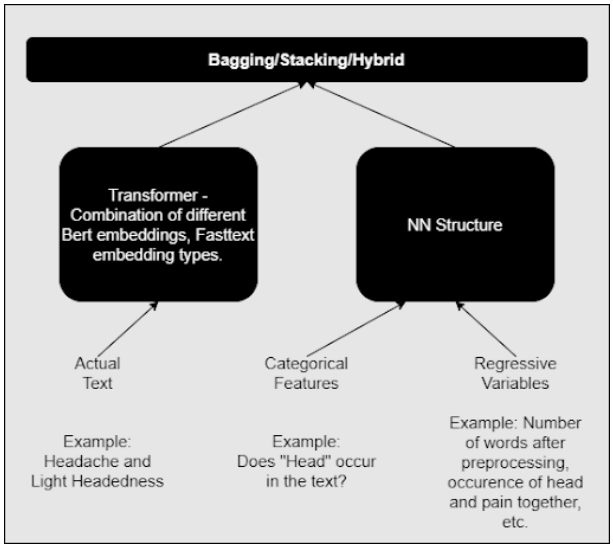

Figure 1: MutiStacked NN model with multiple BERT Embeddings and FastText Embedding

Multiple Transformers based on BERT, FastText are utilized parallelly for the textual data while multiple neural networks are stacked together to determine the categorical features. Please see figure 1 for a visual representation.

3.3.5 NN

The word embeddings were first classified with a vanilla neural network in order to establish some kind of baseline accuracy for the classifier. Using the BERT embeddings, the neural network classified at a 44 percent accuracy.

3.3.6 Fine Tuned BERT With NN

NN with similar structure as above but utilized the attention mask along with pretrained BERT for fine tuning model parameters. This achieved an accuracy of 72 percent. Please see figure 2 for a visual representation.

3.3.7 BiLSTM

With BiLSTM's specialization in the realm of NLP, this model served as a primary target in the model selection process. Using

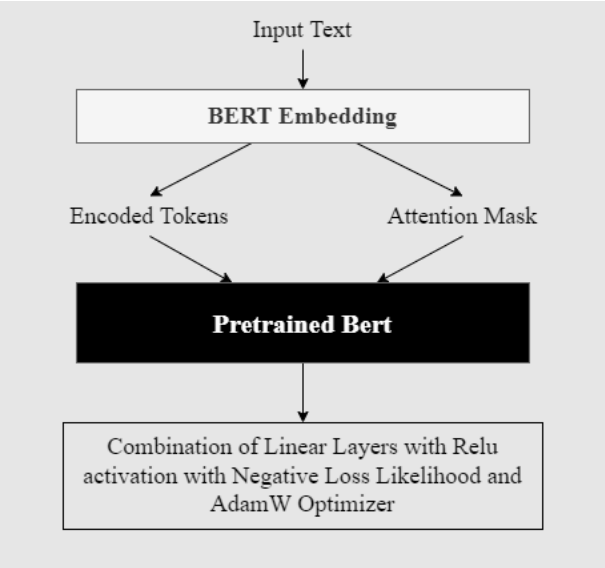

Figure 2: Fine Tuned BERT with NN Model

a similar structure to our previous work in the course, we were able to observe how this model trained on medical data pertaining to ICD codes. Ultimately, through an extensive hyperparameter tuning search, we found a model that classified effectively with 99 percent test accuracy and consistent precision/recall/f1 scores across all classes. The ability to achieve great results emerged from the fact that 100 epochs was possible in a reasonable amount of time, thus allowing for a variety of tuning, specifically with batch size, learning rates, scheduling, momentum, and epochs.

4 Results and Discussion

4.1 Results

Model	Embedding/Encoding/ Features	Accuracy	F1-score
Tree Based Algorithms			
LightGBM Model	BERT + CAT + NUM	98	99
CatBoost Model	BERT + CAT + NUM	96	98
XGBoost Model	BERT + CAT + NUM	98	95
Neural Network Based Algorithm			
Simple NN (no finetuning)	BERT	44	44
NN with BERT FineTuning	FineTuned BERT	72	72
M. Encoding Multimodal	BERT, FastText + CAT + NUM	99	99
BiLSTM	BERT	99.1	99.1

Figure 3: Summary of results for all models

The BLSTM performed well on all significant metrics:

- Test Accuracy: 0.99
- Precision: 0.98
- Recall: 0.99
- F1: 0.99

From the above results, the effectiveness of the BLSTM is observable and consistent across the respective classes. We credit this success with the use of upsampling to deal with significant class imbalance issues, as well as the BERT encoder combined with the structure of the BLSTM.

4.2 Analytic Task and Future scope

4.2.1 Negation

Based on the results we got from the BLSTM model, we confirmed that this is the best model for this type of label classification as supported by other literature. For a brief analysis, the team looked into adding words like "No" or "Not" in the text description to explore the effect of negation in this classification. However, the model did not perform well as it showed a 0.001 percent accuracy in predicting labels. This is a future direction of this project as one can preprocess the data and train the model to be more robust against negation. More data can be acquired by the stakeholders on free text since this project only focused on synoptic text. The problem of negation applies more to free text because usually a doctor's description with a speech-to-text device will include more text such as "no" and "not" whereas the synoptic text is more structured and concise.

4.2.2 Insights on Model Performance

Multimodal Neural Networks are much better than tree based algorithms as it utilises the decision based features that are utilised by the tree based algorithms but additionally also explores the underlying data text relationships using the different transformers.

The architecture of an LSTM model is meant specifically for classifying text data, and therefore is one of the most noteworthy NLP models. Our prior experience in utilizing this model allowed for strong optimism that this would be a useful model to engage with. In addition, the hyperparameters involved in the process (momentum, learning rate, etc.), represent conceptual recognizable concepts to help speed up the tuning process to allow for better results faster. The notion of different gates within the structure of an LSTM allows this model to uniquely perform well on text data, especially as a modified RNN.

4.2.3 Explainability of AI model

In the medical field, the trustworthiness in adopting AI in healthcare is still up for debate. Therefore, further research can be done to present the explainability of the AI models in order to convince medical experts that AI can be reliably used in practice. The better understanding of how the model predicts labels at each step will help physicians to compare and contrast how they make a decision on diagnosing a pathological condition. Our work gives room for advancement in understanding how a model (e.g. BLSTM) predict ICD-O code for a given description in a pathology report.

5 Conclusion

We successfully created a classifier with the BERT encoding and BLSTM structure and optimized the hyperparameters by upsampling and tuning. We also learned that we need to set a cut off point of 10 in order to achieve the best results in classification. The results of this project provide clinicians with a foundation of using AI to predict labels in pathology report. Further research and studies on how negation affects the model and how explainable the model is will be crucial to advance this field and will provide strong evidence for robust ICD-O code classification in healthcare.

References

[1] Heather G., et al. "Assessment of the accuracy of disease coding among patients diagnosed with sarcoma." JAMA oncology 4.9 (2018): 1293-1295.

[2] Barr, Ronald D., Eric J. Holowaty, and Jillian M. Birch. "Classification schemes for tumors diagnosed in adolescents and young adults." (2006): 1425-1430.

[3] Venkataraman, Guhan Ram, et al. "FasTag: Automatic text classification of unstructured medical narratives." PLoS one 15.6 (2020): e0234647.

[4] Wang, Yanshan, et al. "A clinical text classification paradigm using weak supervision and deep representation." BMC medical informatics and decision making 19.1 (2019): 1-13.

[5] Carchiolo, Viincenza, et al. "Medical prescription classification: a NLP-based approach." 2019 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, 2019.

[6] Ollagnier, Anaïs, and Hywel TP Williams. "Text Augmentation Techniques for Clinical Case Classification." CLEF (Working Notes). 2020.

[7] Wang, Ssu-ming, et al. "Using Deep Learning for Automatic Icd-10 Classification from Free-Text Data." European Journal of Biomedical Informatics 16.1 (2020).

[8] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[9] Shaban-Nejad, Arash, Martin Michalowski, and David L. Buckeridge, eds. Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability. Vol. 914. Springer Nature, 2021.