IntelliCheck: A Diagnosis Prediction Model

Arav Parikh and Kaitlyn Bedard
Department of Computer Science and Engineering
University of Connecticut

***Abstract*—Clinical decision making (i.e. the formulation of diagnoses) is a critical part of what doctors do on a day-to-day basis; however, it is a rather demanding task given that it requires them to have an in-depth understanding of disease descriptions and identify hidden patterns among the symptoms. While there is no doubt that these doctors possess this knowledge, the time it takes to recall such information detracts from their ability to spend time and directly work with their patients to achieve better clinical outcomes. With the rise of AI/ML models, though, the automation of the task of diagnosis prediction is not only readily achievable, but perhaps even preferable in the sense that it might allow patients to receive diagnoses without directly consulting a doctor, thereby reducing undue strain on the healthcare system and saving doctor visits for treatments and emergencies. In the following report, we explore past datasets and models used for this, identify areas of augmentation, and describe our chosen implementation and examine the results.**

Introduction

Starting with past models based on the BERT architecture, [1] proposed the BioBERT model as the first domain-specific model in the biomedical sphere. Pre-trained on a large corpus of biomedical data based on PubMed and PMC articles and fine-tuned with several named entity recognition, relation extraction, and question answering data sets, BioBERT outperformed BERT and other state-of-the-art models on the aforementioned NLP tasks in a biomedical context. Given this underlying success with biomedical texts, it makes sense that we might utilize this model and further fine-tune it for the purpose of diagnosis prediction. On that note, though, [2] actually already proposed the Med-BERT model specifically for the task of disease prediction, but the data used to accomplish this task came in the form of electronic health records (EHRs) which are quite different than the typical textual representations passed into transformer models. Although still structured and sequential, EHRs heavily rely on universal clinical code systems used by medical professionals, meaning that both the model inputs and outputs also follow a similar schema. Unfortunately, while undoubtedly very useful in a clinical setting for those who can understand such a schema, it is difficult to see this model being used outside of that setting when more plain-text language (i.e. more easily understood English biomedical terminology) is used, which is exactly what we would like to explore in our own experimentation.

Based on the largely on the BioBERT model, [3] proposed the CORe (Clinical Outcome Representations) model. An important feature of this model is a pre-training step which teaches the model relationships between symptoms, risk factors, and clinical outcomes, in a way that emulates how doctors expand knowledge via experiences and literature. [3] The primary data used in this implementation was de-identified EHR data, in particular, discharge summaries and outcome information associated with a given admission. It was trained using articles as well, including reports from PMC, Wikipedia, and MedQuAd dataset, so unlike the Med-BERT model, the data is more easily understood to those unfamiliar with the clinical setting. The CORe model deals with the following tasks: diagnosis prediction, procedure prediction, in-hospital mortality prediction, and length of stay prediction.

The results of the CORe model are promising. Data in [3] confirms that the model improves the scores on all chosen tasks compared to baseline models, including BioBERT Base and other branches of BioBERT models. There are also indications that the model has learned to predict diagnoses that were never directly mentioned in text. However, despite the promising results, [3] mentions some pitfalls, namely, incomplete/inconsistent labels, the existence of multiple possible outcomes, negation, and numerical data. Issues caused by negation and numerical values can be potentially be combated via semantic encoding, as per the recommendation of the authors. Other issues can be potentially be handled by including more data in the training process.

Project Statement

As indicated in the abstract, we aim to mitigate the strain on the healthcare system while also creating a more user-friendly experience for individuals. Existing solutions, such as searching Google or WebMD, can be misleading and potentially leave users more confused and anxious. Thus, in this report we discuss a diagnosis prediction model which takes as input a simple sentence stating the experienced symptoms and outputs a potential matching medical condition. We experiment with various existing models from the HuggingFace Transformers library in order to leverage the power of transfer learning. We aim to tune these models to predict a diagnosis given a sentence stating the patient's symptoms. Below are links to the models we examine throughout the report:

- Bio-ClinicalBERT
- Monologg/Biobert
- DMIS-Lab/Biobert
- Bioformer-8L
- Microsoft/BiomedNLP

Similar to the models discussed in the literature review, these models are each trained on a biomedical corpus, mainly including EHRs, PubMed abstracts, and PMC articles for many natural language processing tasks such as natural lan-

guage inference, named entity recognition tasks, masked language modeling, and understanding and reasoning.

Data Strategy

Our data collection methodology revolves around aggregating data from three disease-symptom datasets. Each dataset is structured in a way that maps a particular medical condition to a variable number of symptoms. Despite this structure, though, each dataset also presents its own challenge in terms of either cleaning or web scraping the data to eliminate extraneous features and make it viable for our specific use case, but we leave the description of these procedures to the code itself. Below are links to the raw datasets:

- Disease Symptoms — Kaggle
- Disease Symptoms Prediction — Kaggle
- Disease Symptom Knowledge Dataset

Combining the three datasets results in 985 medical conditions in total, of which 931 are unique. The rationale for keeping the repeat medical conditions in the final dataset is because they are typically the more common conditions and, thus, worth having the model learn well. The structure of our collected data ultimately matches that of the raw datasets with medical conditions being matched to a list of their associated symptoms. While we feel as if it would be entirely appropriate to pass this data to the model as is, given the general lack of data in this domain, we instead decide to harness the power of another HuggingFace model to generate more data. This other model turns out to be the paraphraser model – Vamsi Paraphrase Paws. In order to pass our data to this paraphraser, however, we first add the sentence starters "I have..." and "I am experiencing..." to the raw list data so as to form coherent sentences that can be paraphrased. Not only does this technique allow us to generate more data, but it also enables us to better mimic what a user might input when searching for a diagnosis. For each symptom sentence, we generate five more paraphrased sentences, yielding a total of twelve examples per medical condition for a grand total of 11,820 examples in the entire dataset. With such a large number of unique datapoints at our disposal, we are now able to move on to the modeling phase.

Model Strategy

Our model training and testing methodology starts with partitioning the dataset with a 75-15-10 stratified train-eval-test split in order to best assess model performance after training. With this framework established, we then train the five aforementioned models for five epochs each and evaluate them after every epoch in order to form a baseline for comparison. The reason we settle on five epochs for this preliminary training is not only to preserve our limited hardware resources, but also because that is roughly the mark at which each of the models starts to show signs of slowing growth. As a result, for the sake of expediting our experimentation, we assume that the model that performs the best after these five epochs will continue to perform well thereafter. Other important training hyperparameters that we select outside of the number of training epochs include the AdamW optimizer, a learning rate of $2 * 10^{-5}$ with linear decay at a rate of 0.01, and a batch size of 16.

From the baseline comparison of the five models, we determine that the fine-tuning of the Microsoft/BiomedNLP model exhibits the most promising results after five epochs. Hence, we choose this specific model to move forward to the final model training stage of our strategy. Using the same hyperparameters as previously mentioned, we train the chosen model on ten epochs. Further training is determined to be unnecessary as the accuracy shows clear signs of convergence. Thus, the Microsoft/BiomedNLP model, fine-tuned on ten epochs, is chosen as the final model. From this point, we continue to the testing of our model on the remaining 10% of our data.

Results

In Figures 1 and 2, we see the training loss and validation loss, respectively, of each of the prelim models over five epochs. The graphs display similar trends. In both, the Monologg/Biobert model has the lowest loss after one epoch, however, with more epochs, the Microsoft/Biomed model ultimately outperforms the Monologg model. Out of the five prelim models, the two mentioned models are clearly the top performers as they both show a more significant decrease in both training and validation loss over the five epochs than Bio-ClinicalBERT, DMIS-Lab/Biobert, and Bioinformer-8L.

The accuracy results of the prelim models shown in Figure 3 also indicate that Monlogg and Microsoft/BiomedNLP are the best suited models to the task at hand. Again, we see the trend that the Monlogg model begins with the best validation accuracy, but is ultimately surpassed by the Microsoft model. The remaining three models start off with very poor accuracy in the first epoch and are able to improve quickly. However, they seem to converge to a mediocre accuracy of less than 80% each.

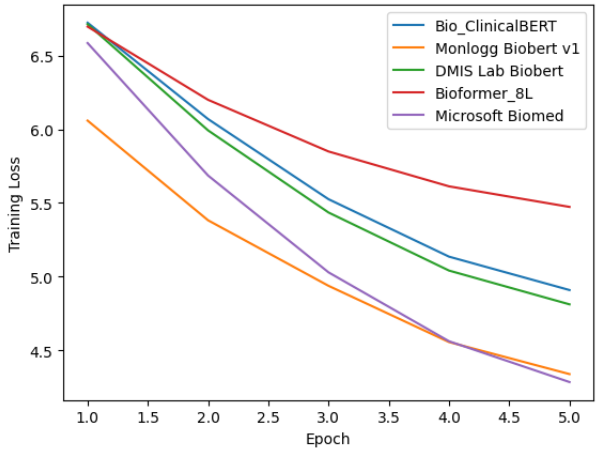

Fig. 1. Training loss of the prelim models

From the above prelim results, the Microsoft/BiomedNLP model was the most favorable model. After all five epochs,

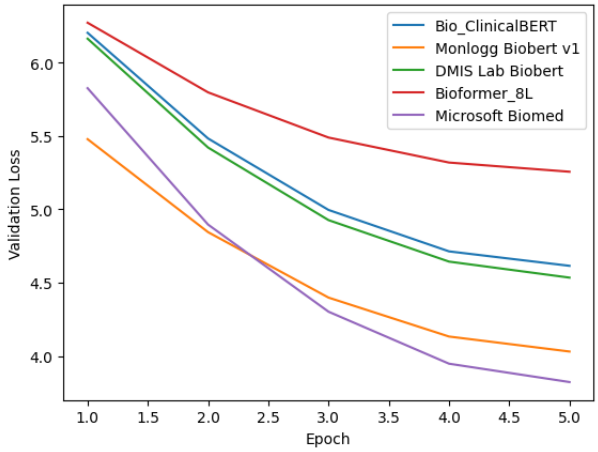

Fig. 2. Validation loss of the prelim models

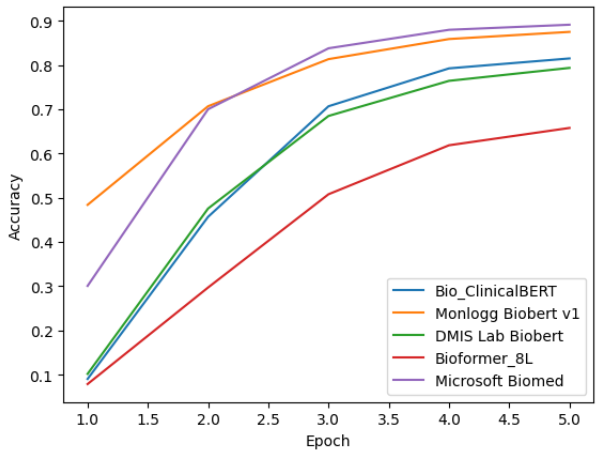

Fig. 3. Validation accuracy of the prelim models

this model had the lowest validation and training loss and the highest accuracy. The chosen Microsoft model also displayed faster learning trends compared to the gradual learning of the next best model, the Monlogg model, as indicated by the orange lines in Figures 1, 2, and 3.

The results of training the Microsoft/BiomedNLP model over ten epochs are show in Figures 4 and 5. In Figure 4, we see a considerable decrease in both validation loss and training loss which indicates that the model is improving with each epoch, as expected. However, there are a few trends that raise some questions, first being that the neither validation or training loss seem to converge. This may be due to the fact that the chosen stopping of ten epochs is too soon for the convergence of loss to be observed. Another question that is raised is regarding the validation loss being consistently lower than the training loss. This may indicate possible flaws in our data or the data split used. For example, the validation data may have insufficient examples compared to the training data.

In Figure 5, however, the model does show convergence in accuracy. After eight epochs the validation accuracy stabilizes at roughly 97%. Hence why ten epochs was chosen as a stopping point. Despite the fact that the model does perform well, given more time and resources, it would be worthwhile to perform further investigations, especially regarding the mentioned concerns with loss in Figure 4.

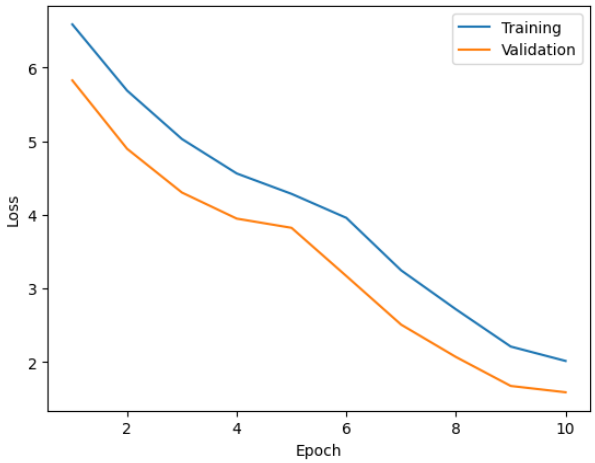

Fig. 4. Loss of the selected model over 10 epochs

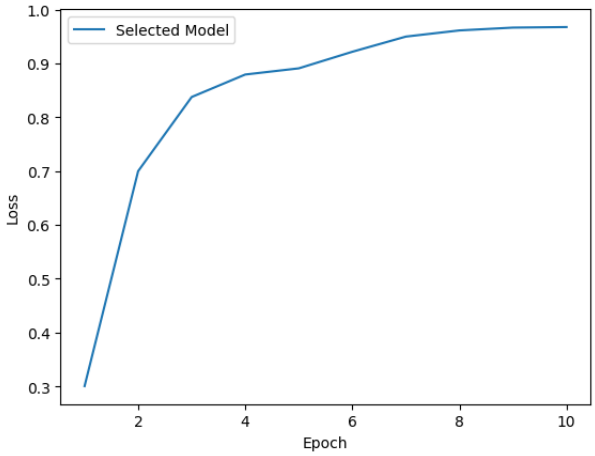

Fig. 5. Validation accuracy of the selected model over 10 epochs

Finally, it is important that we test the model on the withheld data. The results are shown in Figure 6. For the entire test set, we see an overall accuracy of roughly 97%. Essentially, given a list of symptoms, the model is able to predict the diagnosis with high accuracy, being incorrect for only three out of one hundred inquiries. Moreover, for any given symptom set, we see a top three test accuracy of 98.9%, meaning that the correct diagnosis is quite often in the top three predictions outputted by the model. The same appears to hold true for the top five predictions outputted by the model as the top five test accuracy is marginally better at 99.1%. All told, the above metrics all indicate that the model works effectively in making its diagnosis predictions given a simple input of a series of symptoms.

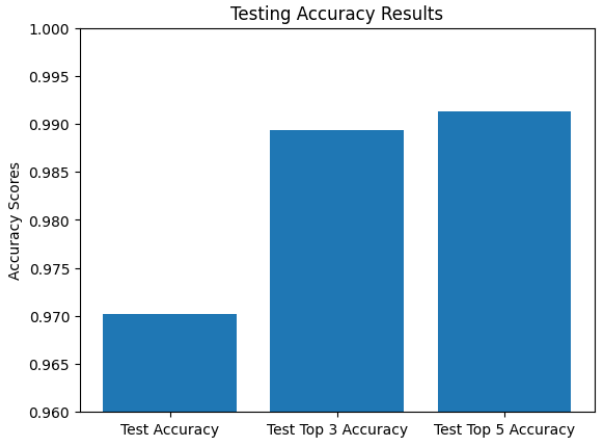

Fig. 6. Testing results of the selected model

Conclusion and Future Work

Although the chosen fine-tuned version of the Microsoft/BiomedNLP model exhibits high accuracy on both the validation and testing data sets, it is important to make note of some potential limitations of our experiment. Firstly, the data is compiled from three data sources, two of which are from Kaggle and the other being from Columbia University. From these datasets, we obtained 931 unique diagnoses which is most likely only a fraction of the existing medical issues in existence. In order to combat the narrow scope predicament, we recommend that any further work seek to include a larger variety of diagnoses. More creative methods such as scraping publicly available and factual medical articles may be a good technique for gathering a more diverse dataset.

Another limitation that arises as a subset of having a relatively small number of diagnoses is the idea of zero-shot or few-shot classification. As the model stands currently, we are unsure of the model's ability to predict diagnoses that are out of its scope, and further evaluation is needed to determine performance of the model on the less frequently observed, or rare, diagnoses in our data.

As for limitations that arise due to the training itself, our current model is not confirmed to have the optimal hyperparameters. Due to the fact that the model demonstrates high accuracy over relatively few epochs, we decide not to perform any tuning. We recommend that future work implement optimization of hyperparameters via methods such as grid search or random search.

Despite the room for improvement indicated by the above limitations, our model still provides promising results that have the capability to make positive impacts on the medical field. With the incorporation of more data, our fine-tuned model has the potential to alleviate extra stress, due to mild or non-emergency requests, on the healthcare system by providing accurate diagnosis predictions. Though it cannot replace the advice and care of licensed professionals in its current state, especially when considering severe circumstances, this model and any further implementations of it can, at the very least, answer lingering questions posed by individuals with minor illnesses at a fraction of the cost. It can also be used by individuals to determine what further steps may be needed in treating their diagnosis or to narrow down possible diagnoses, enabling these individuals to ask their medical professionals the most applicable questions. The many uses of the diagnosis model ultimately seek to aid medical professionals in providing accurate diagnoses in hopes to alleviate or expedite clinical processes.

References

[1] J. Lee et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, Sep. 2019, doi: https://doi.org/10.1093/bioinformatics/btz682.

[2] L. Rasmy et al., "Med-BERT: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction," May 2020, doi: https://doi.org/10.48550/arXiv.2005.12833

[3] B. Aken et al., "Clinical outcome prediction from admissions notes using self-supervised knowledge integration," April 2021, doi: https://aclanthology.org/2021.eacl-main.75.pdf