Improving Patient-Physician Communication in Third-World Hospitals Status Report

Yoonsoo Nam and **Jadrian Tan** and **Alfred Chen** and **Wilson Tan** and **Scott Susanto**
Thomas Lord Department of Computer Science
University of Southern California

1 Tasks that have been performed by then

1. In our original project proposal, we proposed a speech-to-text solution for improving patient-physician communication in hospitals in low-resource settings. However, as noted by our advisor, collecting a human-crafted dataset for this purpose would be challenging. We would either have to expend considerable effort to gather a sufficient amount of data, or we would end up with a small and ineffective dataset. Given these challenges, our current approach involves the technique of knowledge distillation, wherein a more extensive model is employed to train a compact model. This strategy has been chosen keeping in mind the possible limitations and financial constraints associated with accessing state-of-the-art technology in third world countries. By utilizing this method, we aim to create streamlined and efficient models that are perfectly suited for deployment on devices with limited resources. This ensures that our solution is accessible and practical for the healthcare facilities we aim to support.
2. We were able to complete administrative tasks to help with group communication and writing code. Firstly, we have scheduled a weekly meeting time that accommodates our different schedules, and secondly, we have set up a GitHub repository with some base code and data. Since scheduling the weekly meeting time, we have been consistently meeting to discuss challenges and objectives, which has facilitated better collaboration and helped to keep our project on track.
3. Since Knowledge Distillation is a new concept for many of our group members, we allocated time to collectively learn and exchange insights on the topic. This collaborative approach has deepened our understanding of Knowledge Distillation, better equipping us for its implementation in our project.
4.We conducted an extensive literature review to gain a comprehensive understanding of our project domain, as well as to learn from previous works that have been done in this field. This review helped us to identify best practices, common challenges, and innovative solutions that have been proposed or implemented in similar projects, thereby informing our approach and methodology.
5. We have set up the computing environment with the computer we have access to. The owner of the computer will be in charge of running the experiments and the teammates will update the code base through the shared repository in git.
6. We have reviewed the code base for the dataloader provided by the authors of the dataset. Upon consulting with our advisor and confirming that we can use the code base for data loading, we cloned the code base and each member took passes to understand it.

2 Risks and challenges that you think you need to address by the project deadline

1. One challenge for this project is time. We have spent a large portion of our time so far planning our project (pivoting between ideas) and therefore have not written much code.
2. Another challenge for this project is reaching desirable accuracy for the trained models. This includes tuning hyper-parameters, trying different data cleaning/pre-processing methods, or using different training data.
3. As the topic is new to all of the members in the team, having to understand the task while building the code base will be a big challenge.
4. Due to limited compute access, we may run into limitations of how many experiments we can run. This could in turn lead to inefficiency in code base testing as well as hyper-parameter tuning.
5. It is still unclear whether we have enough data to effectively train our model

3 Your plan to mitigate the risks and address the challenges

1. Our main way to combat the time issue is to start early. Although we have not written much code for our project yet, we have a good understanding of our project and are ready to begin writing code to implement it.
2. We will document all of our steps that we took in order to reach different performance baselines. This may include the specific hyper-parameter choices, data cleaning/pre-processing methods, and training data choice. Even if we are not able to in time, we will have extensive documentation that we can report to have others avoid the approaches we have taken.
3. We will allocate a meeting each week to discuss the literature that we have read and check each others' understandings.
4. We understand that this is a problem that all groups must face. We will try to allocate smaller compute to free resources such as Google Colab.
5. If it turns out we do not have enough data to train our model, we can artificially generate more data using ChatGPT as suggested by our advisor

4 Individual Contributions

Due to the shift of project focus, all team members have spent extra time to learn about Knowledge Distillation. Moreover, the topic of using pretrained Transformers is new. Each member have read extensively on different methods used by the available open source models.

4.1 Data cleaning & preprocessing (Wilson)

Wilson has read through the medical data, determined what to remove or replace, and has simplified the data to one file that's easier to visualize. This involved understanding and replacing some abbreviations, removing unhelpful characters, and combining input data with their corresponding output targets.

4.2 Training/Testing (Jadrian, Scott)

Jadrian and Scott implemented a robust data loader specifically designed to handle our training data, which consists of an extensive medical corpus. Their first step was to meticulously create batches of the training data. This involved segmenting the medical corpus into smaller, more manageable units that would be feasible for processing by our machine learning models. Following the batch creation, After creating these batches, the data loader then automates the process of loading these batches into the model for training. This automation is pivotal as it ensures the integrity of the data is preserved while simultaneously optimizing it for efficient processing.

4.3 Computed resource set up (Yoonsoo)

Yoonsoo contributed to setting up the GPU computer to run the code. All GitHub access and updates are completed so that the code can be tested in real time. All relevant packages are downloaded through mini-conda package management. The packages are documented through requirement text file.

4.4 Creating custom models through Huggingface source code (Yoonsoo and Scott)

Yoonsoo and Scott have pulled the source code for Huggingface models to create custom models from it. More specifically, they are testing to create BERT and BART models to test on the dataset to compare the performance baselines.

4.5 Designed model architecture and train/test processes (Yoonsoo and Alfred)

Yoonsoo and Alfred are currently designing the main model that will be using Knowledge Distillation to create smaller models that perform well (close to the previous literature baselines). They are writing the code to train and test Knowledge Distillation on the bigger models.