Karan Sarkar

**Project Proposal**

**Executive Summary**

The federal government offers many services to people with disabilities. One of these is subsidized healthcare costs. As a result, it is in the interest of insurance companies to prove to the government that some of their high cost members are in fact disabled. Insurance companies will be partially reimbursed by federal government for the medical expenses of these disabled members. In reality, the identification of disabled members is quite difficult. Current state of the art models use Tree-Bagger a machine learning model to obtain screen-in rates of around 30% for high confidence members. In this project, we will be exploring natural language processing based methods for disability identification to try and achieve higher screen-in rates.

One problem with current approaches is the so called sparsity problem. There are 70,000 distinct diagnosis codes. However, many of these codes are extremely rare and many codes are very similar. For example, there are codes for breast cancer of the left breast as well as breast cancer of the right breast. However, the current methods treat all diagnosis codes as distinct through the usage of one-hot encoded vectors. This means that we cannot generalize from one diagnosis to several closely related diagnoses. We propose generating a semantic embedding for each diagnosis based on the given text description of the diagnosis. In semantic space, similar diagnoses will have similar representations. A sequence of diagnosis vectors can then be used as input to an LSTM model that predicts disability approval.

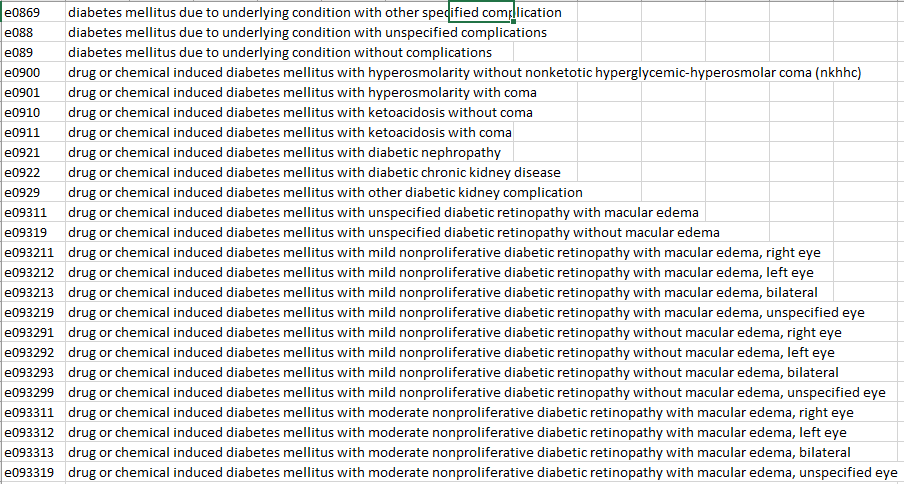
**Goal**

Our goal is to test ways of leveraging the natural language representation of diagnosis codes to create better predictions of disability. We hope this new approach will yield better results than the tree based models that are used currently. We believe that generating semantic representations of diagnoses based on their text descriptions will allow better generalization than the current practice of using one-hot encoded vectors. Thus, another major goal of our project is the creation of robust diagnosis vectors. These vectors could be used in many other applications that use diagnosis codes as input.

**Background and Motivation**

Because of federal subsidies and disability benefits, identifying members as disabled provides a great deal of savings to insurance companies. However, it is very expensive to contact every member out of populations in the millions. Thus, the ability to provide educated guesses as to which members are disabled is very useful. As a result, improving the accuracy of these disability recognition algorithms reduces costs because fewer members need to be contacted.

The current models used are tree-based and use one-hot encoded vectors of diagnoses as input. However, because of memory and processing power constraints only about 2,000 codes out of 70,000 total are considered. These models yield a screen-in rate of about 30%. We propose that the accuracy is low because of this information loss. For example there are many codes related to diabetes mellitus. If the model uses only one diagnosis code for diabetes mellitus, it will be unable to recognize other related codes.



Our project intends to alleviate this problem by utilizing the text descriptions of the diagnosis codes. Similar diagnosis codes have similar descriptions. We will use these text descriptions to generate dense semantic representations of the diagnoses. We hope that now all the different codes pertaining to diabetes mellitus will have similar diagnosis vectors. Thus, we will now be able to generalize to related diagnoses.

**System Architecture**

First, we will generate dense embedded representations of each of the diagnosis codes. In order to do this, we will first obtain a list of members and their associated diagnosis codes. Next, the diagnosis codes will be alphabetized to ensure a consistent ordering and that related codes are adjacent to each other. Note that similar codes have similar alphanumeric descriptions. The list of codes assigned to each members will then be converted into a paragraph where each sentence is a diagnosis description. We now have a list of paragraphs where every paragraph corresponds to a member. We will train the Skip-Thoughts algorithm on these lists of paragraphs to generate the diagnosis embeddings. Second, we will use the diagnosis embeddings to train an LSTM model on the alphabetized sequence of diagnosis codes. We may append additional features like number of claims and average billing amount to the diagnosis embeddings to improve results.

**Time Table**

Skip-Thoughts Diagnosis Embeddings: 3/11-3/18

LSTM Model: 3/18-3/25

Error Analysis: 3/25-31

Final Presentation and Report: 4/1-4/17