Scott Berry – Scorberr- Chatbot Week 13 Assignment 3

**Step 1.** What is your chatbot objective?

* Does it provide a general information (like wikipedia): e.g. what is the tallest mountain?
* Does it have a specific content: e.g. how to fix it?
* Is there any other specific objectives for your chatbot (not mentioned above)?

The medical chatbot aims to answer medical questions from both an average citizen and more technically trained medical professionals. The chatbot should be able to help a user self-diagnose themselves and the according severity of their ailments with recommended courses of action. Additionally, the chatbot may be able to help doctors identify rarer diagnoses from lesser-known symptoms.

**Step 2.** Review and describe your chatbot data structure:

* Do you have user labels? Then you could build a model to train identify users' **Intent**
  + **An “intent” is the intention of the user interacting with a chatbot or the intention behind each message that the chatbot receives from a particular user.**
* Do you have a lot of Name Entities?
* Does your data have emotional content (aka reviews)? If yes, do you have sentiment labels. If not, you could add unsupervised sentiment analysis.
* Do you plan in just mapping user input to tokens in your data?
* What other features do you have (not mentioned above)

The data structure doesn’t initially intend to use user labels, but could flag a user as non-professional or professional based on the language they use in their inputs. User inputs will be mapped to word tokens in order for the chatbot to interpret the words more consistently. The data structure is non-biased medical text of symptoms and diagnoses, there should be no sentiment within the text. The report from my previous presentation highlights the medical package that aids in named entity recognition for medical text in addition.

Step 3. Search online resources (some also provided here as an example)

* Identify several approaches that would fit your data and your chatbot.
* Identify source codes (look for github links to access code) that you could apply to your data
  + How data is processed in these source codes (e.g. word2vec, tf-idf etc)?
  + How data is prepared for training (e.g. train/test split, padding)?
  + What models are used for training?
  + How data is evaluated?

Using resource <https://raw.githubusercontent.com/socd06/private_nlp/master/data/mtsamples.csv>, I found a dataset of transcribed text files from physicians noting client symptoms and giving diagnoses and treatment plans. Using this data, the model uses a word2vec algorithm to assign labels and properties to the text. This data can be prepared by using the Github repository’s 90/10 train/test split. The model used is Syft, a Python library that uses Tensorflow and Grid to learn on medical data that the user does not own. This data is then sorted into 5 fields: Description, Medical Specialty, Sample Name, Transcription, and Keywords.

Step 4. Create a very small subset from your chatbot data and apply source code to train.

* This is just a proof of concept and prototype work - we are selecting aa very small subset to see if the code can be replicated
  + It may not work the first time (you may be missing some dependencies, library versions might be different, code might have changed - but it is ok. You might need look for another source code
* Describe your work (what errors you have encounter etc) and provide references to online resources and source codes

After procuring the notebook provided by the Github repository, I ran the notebook against sample data and was able to get the desired resulting CSV with the 5 fields listed above. By providing the sample text for the chatbot to learn on to the “data” directory, I had no issue as the notebook parsed the data into dataframes and split on a 90/10 train/test split and output resulting symptoms and diagnoses. There were no issues with running this code, given proper installation of dependencies from the requirements file.