Sandeep Guggari
Allison Reibach

Using Machine Learning to Predict Diagnoses from Speech Samples: Intermediate Project Report

I - Introduction:

Primary Progressive Aphasia (PPA) is an umbrella term for neurodegenerative diseases that primarily impact language function. There are three subtypes of PPA, characterized by what parts of the brain are most impacted as well as the nature of language deficit. Each subtype is also caused by a different underlying pathology. For these reasons, getting an accurate diagnosis early on is extremely important, as therapy and drug intervention is going to be different depending on which subtype of PPA a person has. Unfortunately, since this is a rare illness, most clinicians don't have much experience diagnosing PPA. In the early stages of the illness especially, it can be hard to distinguish one subtype from another. The purpose of our project is to take speech samples from individuals with PPA and see if we can use machine learning to diagnose patients with the correct subtype. We think a program like this could be a useful tool to clinicians.

II - Algorithms, Rules, and Data Structures:

Our main approach is to implement a support vector classification model (SVC) that receives numerous data points extracted from our transcribed speech samples, each capturing different linguistic features. From our research, we have found that SVC works particularly well when faced with multidimensional data.

The speech samples we are working with are transcribed .cha file types. Some useful pieces of data are automatically extracted from the .cha file and placed into a .csv file. Among these are the number of words per utterance, the number of morphemes per utterance, the number of times a participant repeats themselves, and the percent of words that are any given part of speech. We wrote a quick script that calculates the mean length of words for each speech sample and added this data to the csv file.

The next thing we wanted to do was train a Naive Bayes Classifier to predict the variant of PPA based on part of speech transitional probabilities, and include its output in our SVC model. To do this, we tagged each word in the .cha files with parts of speech using the tagging function of the NLTK library. We then generated an array of isolated POS tags in sequential order, and converted these arrays into trigrams so that they are all the same length, which was a requirement of our Naive Bayes classifier.

We fed these part of speech trigrams into a Naive Bayes classifier to see if PPA subtype can be determined from transitional probabilities of part of speech tags using the sklearn library. The sklearn library is an open-source machine learning library that has an abundance of features and models tuned for various situations.

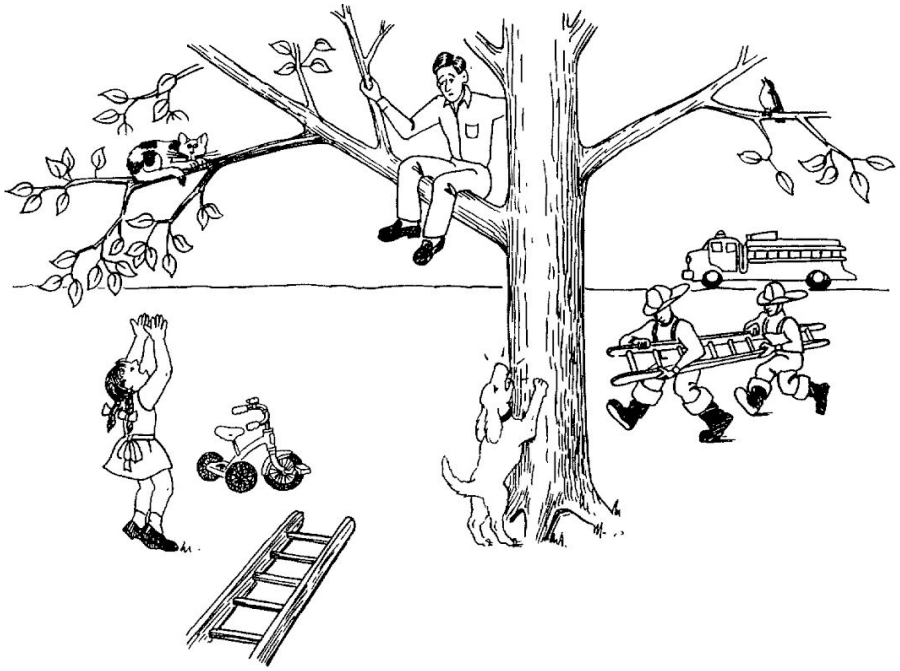
The motivation for using this part of speech tag approach resides in the fact that in some forms of Primary Progressive Aphasia, patients are prone to omit function words (ie: “tree green” as opposed to “the tree is green”, where “the” and “is” are function words). We are hoping that our model will be able to pick up on other more subtle differences between how patients with different subtypes of PPA structure their sentences.

Like we mentioned above, the result from Naive Bayes classifier is not to be taken as the final prediction. Rather, we plan to integrate the results from this model with the SCV model add it as a data point in our .csv file. Then we can perform a fitting of the SVC model and hopefully make more accurate predictions than before.

One of the libraries we used was called Pandas. This library allowed us to import .csv files into python in the form of a data structure called DataFrames. These DataFrames function as multidimensional arrays that allow for easy data parsing and visualization. We also use the sklearn library as mentioned previously. Other data structures we used included lists and dictionaries. The dictionaries were key in determining part of speech counts for use with the Naive Bayes classifiers. Lists were used to hold part of speech and trigram (determined using NLTK) sequences.

III - Corpus Resources:

We are using transcribed speech samples from PPA patients collected from Dr. Henry’s Aphasia Lab. In the speech samples, the patients are instructed to describe the picture below:



The transcriptions are done by hand by listening to a recording of the speech sample, and they include codes for certain types of errors. They are saved as .cha files, and an example output from a patient with non-fluent PPA can be found below:

*SE006: &uh the &um cat's in the tree and the &um &uh cat's in the tree and the &uh &uh little girl is trying to get it out .
*SE006: and &um &uh the &uh cat's in the tree and the little girl is trying to get out and &uh so the maybe the father is going to get the cat .
*SE006: and &um &uh &uh it so the fire department is coming around and the &um supposedly getting the cat out of the tree .
*SE006: and &um so the firetruck and &uh the &uh firetruck in the +...
*SE006: so the dog is &um getting the &uh
*SE006: the dog's barking .
*SE006: and &uh the &um fire ya know <the ladder> [//] <hook in ladder> and &um &uh the hook in ladder is the +...
SE006: &uh I think the &uh person the father is talkɪn@u [:talking][p][*p:n] the tree .
*SE006: and the cat &uh the cat is &uh +...
*SE006: so the firemen are going to get the cat out of the tree .
*SE006: and the &uh dog is barking .
*SE006: and &uh the &uh the father is &uh stuck in the tree .
@End

Although this text may appear difficult to work with, there is a python package called Pylangacq designed for working with these .cha files. This allows us to derive as much useful information as we can from the transcriptions as they come, without having to reinvent the wheel.

IV - Initial Results

We were able to get the SVC to work with limited results. So far, it's only 40% accurate, but this is something we hope to continue developing as the semester does on. We will get a larger sample size, add dimensions that will help us to find meaningful differences, and remove dimensions to avoid overfitting and increase generalizability.

In addition, our Sklearn Naive Bayes Classifier is making predictions with an accuracy of 39.6%. This obviously isn't great, but we intend to write a script to include START and END tags at the beginning and end of every utterance. We think that this is necessary for the classifier to have the most useful data from our transcriptions as possible. We also intend to add more training data and more testing data. Currently, our training data set is 11 transcribed picture descriptions, and our testing data is 3 transcribed picture descriptions. We hope to expand this considerably and also include transcribed picture descriptions from age-matched controls, healthy adults who do not have aphasia.

V - Contributions

Alli - Worked on collecting and organizing data from the lab, the POS tagger, a short program to calculate the mean word length within each transcription files, and the Naive Bayes Classifier.

Sandeep - Worked on importing the .cha files, the Naive Bayes Classifier, and the Support Vector Classifier.

VI - References

1. Information on the different variants of Primary Progressive Aphasia, and current diagnostic processes
   Gorno-Tempini, M.L. et al. "Classification of Primary Progressive Aphasia and Its Variants." *Neurology* 76.11 (2011): 1006–1014. *PMC*. Web. 21 Mar. 2018.
2. A paper that uses similar computation techniques to attempt to differentiate between two of the three subtypes of PPA and control speech samples. A major difference between our project and this paper is that we are testing for part of speech transitional probabilities, and we are including all three variants of PPA.
   Fraser KC, et al., Automated classification of primary progressive aphasia subtypes from narrative speech transcripts, Cortex (2013), http://dx.doi.org/10.1016/j.cortex.2012.12.006
3. Pylangacq toolbox
   Jackson L. Lee, Ross Burkholder, Gallagher B. Flinn, and Emily R. Coppess. 2016. Working with CHAT transcripts in Python. Technical report TR-2016-02, Department of Computer Science, University of Chicago.
4. Sklearn library, used for Naive Bayes and SVC
   Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
5. NLTK
   Steven Bird, Ewan Klein, and Edward Loper (2009). Natural Language Processing with Python. O'Reilly Media Inc. http://nltk.org/book