NLP Final Project Proposal

Yuval Timen

For our NLP final project, we will be evaluating the performance of multiple classifiers and feature extraction methods on the task of Authorship Attribution.

We have gathered the complete works of a few notable authors, listed below with links to the data:

- Aristotle
- William Shakespeare
- Henry Wadsworth Longfellow
- Percy Bysshe Shelley
- Winston Churchill (the American author, not the British PM)
- Robert Browning
- Friedrich Nietzsche

There may be additional authors' works included in this list, pending the analysis of the data. In order to ensure that the task is not trivial, we want to ensure that there is a decent overlap in the vocabulary between datasets - otherwise, the classifier should be able to use the presence of particular words to determine authorship. This would be a more trivial example, and we wish to avoid such easy classification.

The tools we will be using include:

- NumPy and Pandas for data manipulation
- NLTK for data cleaning/preprocessing
- ScikitLearn for models and other Machine Learning functionality
- Matplotlib for graphs and visualization

We begin with the data analysis/exploration section. First, we examine some basic statistics about each data set:

- Number of sentences
- Number of words
- Average sentence length
- Average word length
- Vocabulary size

Next, we examine the differences in vocabulary between these our authors' works:

- Run a pairwise set-difference on the vocabulary
- Use Matplotlib to visualize vocabulary differences

At this point, we decide whether to add additional authors' data. If there is a high enough similarity between vocabularies for enough authors, then we will continue, otherwise, we will substitute the more unique authors' works for works with a higher similarity to the ones we have.

Next, we will run a cascade of data cleaning and preprocessing. This will include the use of NLTK's sentence tokenizer and RegEx substitutions. Our final data file for each author will be a csv file containing 2 columns: class (signaling the author's name) and data (one cleaned sentence from author's corpus). We will then shuffle the data and conduct an 80-20 train-test split.

For feature extraction, we examine three main methods:

1. One-hot encoding of the words in the sentence
2. TF-IDF encoding of the words in the sentence
3. The best cluster found from running K-means

We are now ready to create the classifiers. Our baseline will be a Multinomial Logistic Regression classifier. In addition to our baseline method, we will use a Feed-Forward Neural Network. For the Neural Network, we create A different Neural Language Models, each trained with a different author's data (where A is the number of authors). The classification is done by evaluating the score of each model and taking the argmax over all language models - the classification will be the author whose Neural Language Model produced the highest likelihood.

At this point, we will tune some hyper parameters of our model, and once we have attained a desirable set of hyperparameters, we proceed to the next step of evaluating our models.

For the evaluation step, we will be assessing our baseline Logistic Regression model against our Neural Language Model with the usual metrics: accuracy, precision, recall, and F1 score. We then use both the Logistic Regression model and the Neural Language model as our new baseline, in order to test the efficacy of different feature extraction methods, such as TF-IDF vectors and K-means clustering. We evaluate these differences in the same way as we evaluated our models.

Finally, we present our conclusions. We include an overall summary of our experiment, as well as our findings and evaluations.

By December 2nd, we plan to have all of the data imported and the models implemented. At that point, we will be working on producing visualizations, documenting our code, and debugging any pesky bits of the experiment.

SOURCES:

1. https://www.quora.com/LDA-Topic-Modelling-output-what-do-the-output-values-represent
2. https://www.aclweb.org/anthology/C18-1029
3. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.2951&rep=rep1&type=pdf
4. https://scikit-learn.org/stable/modules/feature_selection.html
5. https://towardsdatascience.com/hyperparameter-tuning-c5619e7e6624
6. https://www.aclweb.org/anthology/C18-1029.pdf
7. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.2951&rep=rep1&type=pdf
8. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7256385/
9. https://towardsdatascience.com/kmeans-clustering-for-classification-74b992405d0a#:~:text=KMeans%20is%20a%20clustering%20algorithm,than%20the%20number%20of%20classes.