Will Morgan Final Capstone Proposal January 2019

I am planning on solving the problem of classifying texts based on their subject. Solutions to this problem are valuable because they allow businesses or any other entity to organize large amounts of text. This common for an info email inbox or maybe a help desk. I plan on using data that I scrape myself from the internet. Specifically, I plan on scraping the closed captioning of mathematics lectures from youtube. Why the closed captioning? Because it offers an opportunity to learn new APIs (which is how I plan on accessing the data) and other tools to obtain custom data. Why math? I chose math because all the subjects are generally related but nuanced enough to have their own distinctions. Also, because I enjoy math. This should be the most challenging part of the project, specifically, getting the subtitle files and extracting the text programmatically. There aren’t really any tutorials for this specific task.

Techniques from the course I plan on using:

I will refer to the scraping section of the curriculum to see if it is relevant for my objective. Other techniques include TF-IDF vectorization, clustering algorithms, silhouette scores, cosine similarity and the suite of supervised modeling tools from sklearn.

Other techniques that I plan on using not in the curriculum: Encoding the text to integers, reducing the dimensionality with t-SNE, using Doc2Vec.

Techniques I don’t plan on using: spacy, NMF, LDA. I have a previous project with extensive work in this area (NMF, LDA). In addition, I don’t like the way spacy handles text.

The idea is to compare Doc2Vec vs TF-IDF using three key metrics. First, how well does the data cluster? Second, what is the overall and pairwise similarity for each lecture. Third, which vector representations are the best for supervised classification?

How do these metrics impact the choice of the model? An example of a business need paired with the model choice is given for each metric.

1. A business has an unattended [info@someco.com](mailto:info@someco.com) inbox. The manager wants to determine how they can categorize these emails. Which ever model is better at clustering would be chosen to assist in determining how many different categories would be needed to efficiently organize each message.
2. A business wants to be able to recommend products to their customers they are likely to buy. The cosine similarity matrix would be a foundation for recommending new products based on cosine similarity. This is no different with text. If a customer likes a certain lecture, we want to be able to recommend other lectures with high similarity. In this case would select the model that gives a higher overall similarity.
3. A business has a continuously functioning helpdesk and must dispatch support. A classification model could be used to assign help tickets (based on textual content) to the most relevant technician (or handler) for that ticket. In this case the more accurate classifier could be used to assign the handler.

These applications allow businesses to function more efficiently by reducing the time and resources that would otherwise be required to accomplish the objective.