Melissa Melnick
Homework 3
IEMS 308
03/07/21

Forewarning:
This project had some severe limitations, the biggest of which was the sheer computing power of the machine it was being run on. The remainder of this report will detail the steps that should be taken to create this system. The code provided in theory should also create the desired system, but is commented out due to the lack of computing power of the machine.

Methodology
The overall task for this project was to create a system that could select the answers from a corpus for the following four questions:

1. Which companies went bankrupt in month x of year y?
    a. Example: Which companies went bankrupt in September of 2017
       Answer: Toys R US
2. What affects GDP?
3. What percentage of drop or increase is associated with this property?
4. Who is the CEO of company X?
    a. Example: Who is the CEO of Facebook? Answer: Mark Zuckerberg

First, the text files were read into Jupyter Notebook. The articles for 2013 and 2014 were held in separate files, so each group was read in separately, then combined. Next, using the NLTK package each article was broken up into its individual tokens, and a new set was created of all the unique tokens present in the corpus.

The NLTK package was also used to tokenize each sentence in the corpus, and then from there each tokenized sentence was also word tokenized. While this may seem a little bit redundant, storing the word tokens in these two individual ways is necessary for some of the following steps.

Following this extraction process, a bag of words was created. This was done by creating an individual dictionary for each sentence in the corpus, each containing the words and their frequencies found in that sentence. This bag of words supplies the main foundation for the Document-Term matrix created shortly.

Before creating the Document-Term matrix, the TF-IDF scores had to be calculated for each word in a given document. A TF-IDF score is made up of two components: Term Frequency (TF) and Inverse Document Frequency (IDF). TF is a measure of how often a given term appears in a document. Likewise, IDF is a measure of how many documents out of the corpus the given word appears in. IDF is a very good measure of term importance within the corpos.

Once these scores were calculated, it was time to build the Document-Term Matrix. We created an initial DataFrame of all of the TF-IDF scores. From there, we converted all of the sentences in the corpus to TF-IDF vectors using the Sklearn Feature Extraction tool. After converting all of the sentences into these vectors, we made those into a new DataFrame - our document term matrix!

Now that we have our Document-Term matrix, we can start analyzing how we could pick the best candidate to answer a given question. For this, we will need to find all of the key words in the question being asked. We did this using the Spacy package. Spacy is an excelling tool for things such as Part-of-Speech tagging and Named Entity Recognition. In this case we used part-of-speech tagging. Any word in the question that was not labeled as a ‘stop words’ or an extremely common and typically uninformative word, or as punctuation were taken as key words.

Next with our Document Term Matrix, we are able to create a score for each sentence in the corpus, and how well it matches the question. This score is made up of three components:

1. The number of keywords that match words in the sentence to be scored
2. The sum of the TF-IDF scores for those matching keywords
3. The sum of the TF-IDF scores of the keywords that were not found in the sentence to be scores

The final score can be calculated as follows:

Final Score = # of matching Keywords + TF-IDF sum of matched keywords - TF-IDF sum of unmatched keywords

Once these scores are calculated we can sort through our scores to find the sentence with the highest overall score.

After we have chosen our sentence, we must figure out how to extract the desired answer from that sentence. Since we have four different questions, we labeled each of them as a different type of question class. Based on the question class, we can search for different things within our chosen sentence.

For the question about bankruptcy, we can use Spacy NER to pull out the name of the organization mentioned by using the “ORG” label. For the factors that affect GDP, we can use Part of Speech tagging to pull out the Nouns. For the percentage question, we can pull out the numbers. And for the CEO question, we can use the “PERSON” label from spacy.

It is important to note that we create models to pull out CEOs, Companies, and Percentages in the last iteration of this project. However, these models did not perform nearly as well as we had hoped, so in creating this model, we opted not to use them.

Overall Findings

Obviously the lack of computing power on our machine was a huge hindrance to this project. There is unfortunately no way to be certain that our model would correctly extract correct answers from our corpus, but based on some human observation, there are some interesting things to make note of. The final step: the extraction step, is certainly not specific enough. For example, the “PERSON” label that Spacy provides is almost certainly not enough to correctly identify a CEO, especially if there are multiple people mentioned within a sentence. This same theory/issue is most likely present within all four extraction algorithms. Ideally, with the addition of working models from the last iteration of them project, we would be able to apply them and achieve more accurate results.