NAACP Spark Project Final Report

Gowtham Aosokan,Temi Lajumoke, Ningxiao Tang, Hong Xin

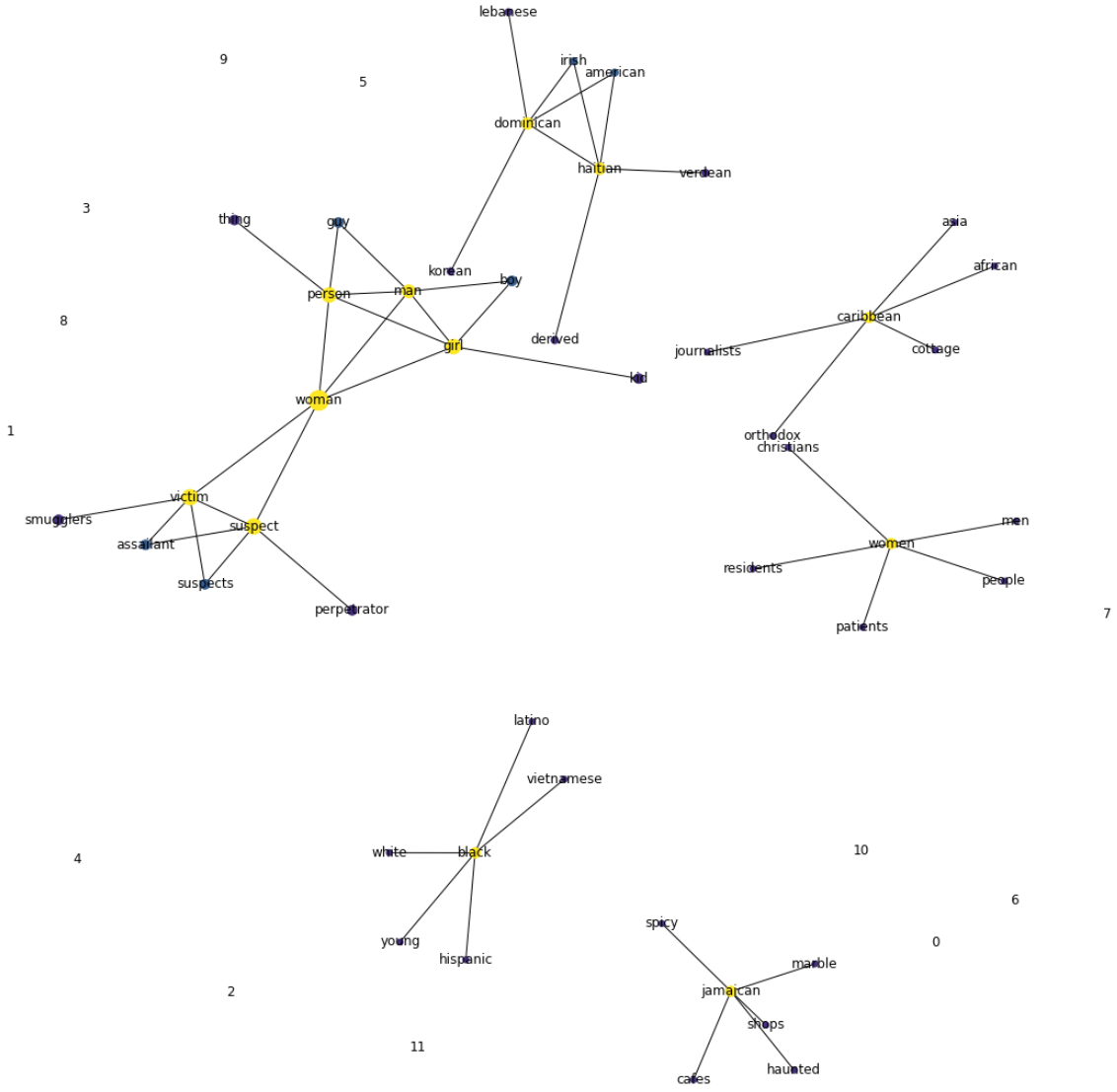
## Topic modeling

We implemented a topic modeling method using the Doc2Vec model. Doc2Vec is a technique for natural language processing. It uses a neural network model to learn document associations from multiple documents, such as articles or reports. In this project, we use Doc2Vec to identify top-k similar words for a list of target words and create a word cloud and a wordnet visualizations. Doc2Vec is very similar to Word2Vec. The differences are Word2Vec is on the words level and Doc2Vec is on the documents level.

## Doc2Vec & WordCloud: Code Explanation

1. We first used pandas to import and merge data from Boston Globe between 2014 to 2018. Then, we removed special characters from data.
2. Next, we tokenized the text using the gensim package, and created tagged documents based on the tokenized text.
3. Next, we initialized the Doc2Vec model using the tagged documents. Given a list of target words, such as words representing races and ethnicities, we pick random words from the target words to run our model.
4. We can infer a vector from the Dco2Vec model and thus find their corresponding similar words. Using the networkx package, we created the wordnet visualization. An example graph for Boston Globe 2014 is shown on the next page.

## Graph




1. Yellow circles represent the target words that we picked.
2. Purple circles represent the words that appear frequently with a target word.
3. Blue green circles represent the words that appear frequently with 2 target words.
4. The size of each circle represents the frequency of the word. For example, a bigger circle means the word appears more frequently.
5. As we are showing in the picture, not all words in the purple and Blue green circle have positive or negative meaning. A next step that we can do is to further categorize and analyze those words.

**BERT Model & Transformers**

BERT, the Bidirectional Encoder Representations from Transformers is a language representation model designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As such, we can fine tune already pre-trained transformer models (BERT, XLM-RoBERTa, ALBERT, etc.) to solve a wide variety of problems without any major architectural modifications.
The BERT model processes each token of input text in the full context of all tokens before and after, using computed vector-space representations of these tokens.

**High Level Code walk through and explanation.**

We use BERT for one major task: sentiment analysis for articles of the Boston Globe, but we take that a step further using word co-occurrence to determine which sentences in what articles we'd like to determine sentiment for.

**Sentiment Analysis**

For the sentiment analysis tasks, we finetune the BERT model for sentiment classification with the Large Movie Review Dataset that contains the text of 50,000 movie reviews from the Internet Movie Database (IMDB). The IMDB dataset contains a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. We use 20% of this training dataset for validation, and the other 80% for training. We also use all 100% of the test dataset for testing.

We selected the BERT english uncased model (bert_en_uncased_L-12) with 12 Transformer blocks as the BERT (base) model of choice from the TensorFlow hub. The model architecture is described as follows:
Our model contained an input layer, followed by a preprocessing layer, to preprocess plain text inputs into the input format expected by BERT, next had passed preprocessed input into the BERT model (transfer learning) and selected the 'pooled_output' layer. Next we had a dropout layer for regularization, followed by a Dense layer for classification.

We trained multiple models with different hyper parameters. Epochs ranging from 2-5 epochs (BERT recommends we stay within 2 and 4), and learning rates from 2e5 to 5e5. We also used the adam optimizer, as well as the Binary cross entropy loss (since this is a Binary classification task). Ultimately, this model produced a  zero-shot accuracy of 84.8%.

**Relative occurrence of words: (link)**

Building upon the sentiment analysis task, we used the relative occurrence of words to determine articles with mentions of predetermined race or ethnicity. We did this by selecting articles that contained sentences with a relative distance or span between selected words.  This approach provided a much better way to capture articles with sentences where the anchor word (usually an adjective like “Jamaican”), as well as the selected noun (“woman”), were mentioned, even in cases where they were not positioned next to each other in a sentence.

**Future work**

At the moment, the BERT model is finetuned for sentiment classification with the Large Movie Review Dataset from the Internet Movie Database (IMDB). We plan to have it finetuned instead with labeled data from Lexis Nexis.

We also plan on improving how we determine explicit mentions of race and the context they are used in sentences.