Fall 2018

Natural Language Processing (NLP) of Liberal Arts College Newspapers in Ohio over 30 years

Shane Canfield
Kenyon College, canfield1@kenyon.edu

Natural Language Processing (NLP) of Liberal Arts College Newspapers in Ohio over 30 years

Shane Canfield, IPHS/CWL 391 - Artificial Intelligence for the Humanities

Professor Chun and Professor Elkins

INTRODUCTION

Computers have been extremely useful in humanity's quest for knowledge, performing calculations and other strenuous tasks in seconds. For a computer to perform the tasks, it requires a specific set of instructions, or code, to tell it what to do. These series of commands and instructions are strict, in that any syntactic error results in faulty, or zero functionality.

Human language is very much unlike that of a computer, in that it can be grammatically incorrect, irregular, or even incomplete, yet another human may still get the point and understand the information being exchanged. A significant part about what makes us human is the ability to use and develop our dynamic language. When a computer is able to completely understand and mimic human language, we will likely have something closer to artificial intelligence than anything we've seen yet. Development in this area has lead to Natural Language Processing, or NLP.

On March 31st 2017, students, professors, and hobbyists alike gathered together at the HackOH5 Student Newspaper Hackathon to analyze 170,000+ pages of student newspapers from 5 colleges: Kenyon, Denison, Oberlin, Ohio Wesleyan, and the College of Wooster. Spanning over 160 years, the digitized libraries were filled with years of student coverage organized in a huge dataset of text and images. What can be done with all of this newly digitized information?

NLP allows us to analyze, visualize, and contextualize this textual data. This project aims to analyze textual information recorded between 1970 and 2000 by three different colleges: Kenyon, Denison, and Oberlin. By using NLP, any word used by any school from any issue can be mapped to an n-dimensional semantic space where the distances between the words can be used to represent their semantic closeness. For example, words like student will be closely associated with professor, college, people, alumni, etc. By investigating specific words that are historically relevant, we can try to understand how each college might perceive certain events and compare them with each other.

METHODS

Jupyter Lab

Jupyter Lab is the newest interactive development environment (IDE) from Project Jupyter. 100% open source, free for all software that allows users to create a web-based environment for creating Jupyter notebook documents. These Notebooks contain an ordered list of input/output cells which can contain anything from code and other text to mathematical models and plots. It was within the Jupyter Lab that I executed all of my commands and structured the data.

Gensim and word2vec

Gensim is a popular open-source natural language processing library. It uses top academic models to preform complex tasks like building document or word vectors, corpora, and performing topic identification and comparison. Gensim splits the text files into different lines, and Word2vec creates vectors of words taken from these lines. Working together, these packages help represent words in the n-dimensional space.

The first thing to do was structure the massive data files that were provided by HackOH5. The digitized newspapers were in XML format, so in order to be able to work with them they had to be converted into comma-separated values, or .csv's. The XML files were first converted into text files, and separated based on school, and then separated again based on decade. I decided to focus on years 1970-2000.

In the code cell below, the denison_70.csv file is first opened, and then it's text extracted and split into different lines. Then, each line is split into lists of strings, where each string will represent a word.

```
[2]: import pandas as pd
df = pd.read_csv('/Users/humanities/Desktop/Hackathon/denison_70.csv')
corpus_text = '\n'.join(df[:50000]['content'])
sentences = corpus_text.split('\n')
sentences = [line.lower().split(' ') for line in sentences]

[3]: def clean(s):
    return [w.strip(',."!?:;()\'') for w in s]
sentences = [clean(s) for s in sentences if len(s) > 0]
```

Gensim will now be used to create the model that will help predict the semantic clustering around words of our choosing.

```
[4]: from gensim.models import Word2Vec

model = Word2Vec(sentences, size=100, window=5, min_count=3, workers=4)

[5]: vectors = model.wv
```

Now that we have the vectors, we can begin to search for specific words and find the words that are most semantically similar. These words can be searched individually, or we can also calculate the similarity between two specific words, given as an output between 0-1 (1 being 1:1 in similarity)

```
[32]: vectors.most_similar('school')

[32]: [('graduate', 0.752156674861908),
       ('career', 0.6943222284317017),
       ('high', 0.6708678603172302),
       ('schools', 0.6530600786209106),
       ('standing', 0.6403533816337585),
       ('college', 0.621809720993042),
       ('law', 0.5911512970924377),
       ('job', 0.5902386903762817),
       ('elementary', 0.5874952673912048),
       ('prep', 0.5864046216011047)]

[34]: print(vectors.similarity('team', 'loss'))

      print(vectors.similarity('kenyon','victory'))

      0.708019712647016
      0.7797207616538336
```

RESULTS

Now that we understand the way this model works, I wanted to see if it was possible to contextualize some interesting similarities with events during that time period. I chose three events within 1970-2000 (President Nixon resigning, the Berlin Wall coming down, and the re-election of President Bill Clinton) and used key words that I think would be able to tell a story.

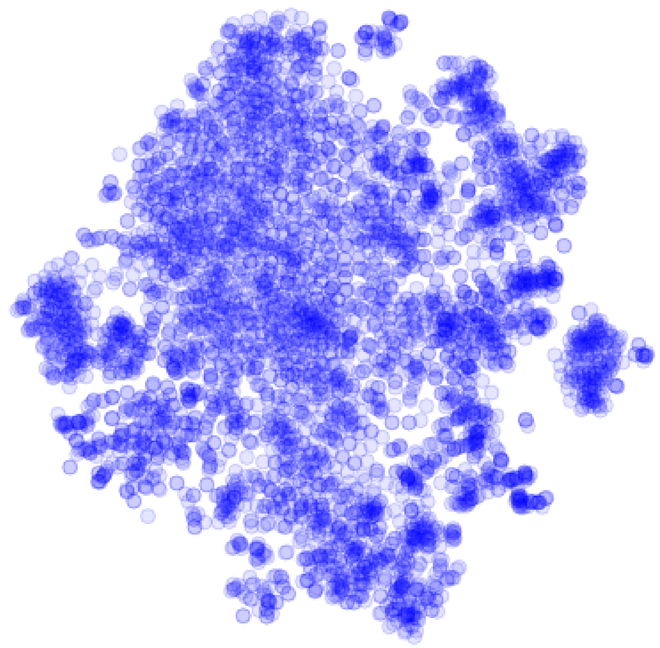
	Similarity Score (0-1)		
1970s	**KENYON**	**DENISON**	**OBERLIN**
"Nixon" and "Corrupt"	**0.54**	**0.24**	**0.36**
1980s			
"Berlin" and "Victory"	**0.12**	**0.42**	**0.04**
1990s			
"Clinton" and "Help"	**0.18**	**0.06**	**0.04**

Looking at the similarity values outputted by the algorithm. A few things are apparent.

1) Kenyon seemed to be more inclined to put "Nixon" and "Corrupt" within the same sentence. Kenyon might have had stronger reporting of the Watergate scandal.
2) Denison reported the words "Berlin" and "Victory" more frequently together than any other school during the 1980s, obviously referring to the Falling of the Berlin Wall.. Oberlin barely seemed to report on it, or perhaps had a different kind of discussion.
3) Kenyon might have liked President Clinton a little more than Denison and Oberlin, considering it was 3 times more likely than Denison to cluster "Clinton" and "Help" together

t-SNE Word Embeddings

t-SNE is a a technique that allows visualizations of similarity between objects, or in this case, words. It takes a group of high-dimensional vocabulary word feature vectors, and compresses them down to a 2-dimensional (x, y) coordinate. Words that score high levels of similarity will be near each other in this figure below, and words that are least similar, will be furthest away. The figure below represents the t-SNE visualization of Kenyon's data from 1970. With more data, this would be more of an accurate representation of word choice used by each of the schools.



CONCLUSIONS

It is interesting to see how different schools react to events in history, and by using Gensim and word2vec, it is even more intriguing to be able to find this out with a simple word search or word comparison. Using this technology to map words in a semantic space is confusing technology, but the more work that is done to clean and curate the data, the clearer the resulting picture will be. The highest similarity score I was able to achieve was a 0.54 in Kenyon's Newspaper with the words "Nixon" and "Corrupt". This is likely due to the fact that the political scandal was given more airtime on the media than the rest of the events chosen. Gensim with word2vec works, and it will be interesting to see the types of technological capabilities that will come from advancements in this field.

OUTLOOK

There is always more work that could be done to make these analyses more accurate, however there are definitely some interesting paths that could be explored

- While cleaning the data, there were some typos and obvious concatenation of words that would be recognized as an entirely new word. Getting these words into the data would help the accuracy.
- Investigate secondary effects like changes in newspaper editors, authors of stories, and other content choices that would potentially skew that data.
- Expanding the timeframe to include all 170,000+ pages, and further explore changes in college sentiment towards certain topics over time.
- Expanding the data set to include all Ohio 5 Colleges, and further explore the differences between the college's views over time and contextualize them based on real-world events

SOURCES

Code derived from Emmanuel from Kaggle:
https://www.kaggle.com/rbhambri/word-embeddings-with-gensim-visualistion/notebook

ACKNOWLEDGEMENTS

I would like to thank Professor Chun for his support and time spent helping me on this project.