– Independent Work Report Fall 2018 –

Textual Analysis of Daily Princetonian Archives

Yang Song

Adviser: Prof. Brian Kernighan

Abstract

This project studies Daily Princetonian archives from 1946 to 2015, quantitatively analyzing the text within articles to examine Princeton's history through a linguistic lens. Using several word embedding models including Word2Vec, this project explores the relationships between words, capturing word similarity and association, as well as the change in usage of words over time. Visualization tools of relative frequency counts for words and phrases are also created. This project extends several earlier projects that perform textual analysis of the Daily Princetonian archives which instead focus on n-grams and sentence-level sentiment analysis, by using a word vector approach to analyze the relationships between individual words from a Princeton perspective. Using word vectors allows us to determine analogies between words, as well as semantic similarity. We examine several natural language processing models and compare their effectiveness for textual analysis and data visualization. This project builds upon efforts to digitize the Daily Princetonian archive through applying modern tools in natural language processing to analyze the text from articles and visualize data describing trends throughout Princeton's history. Through focusing on newspapers from the last seven decades, we can observe how major events, including the admittance of women to Princeton's campus, protests against the Vietnam War and changes to Princeton's Eating Club admittance are represented through language. The results and visualizations reflect Princeton's cultural, historical and linguistic change throughout time and can be used by future scholars, researchers and historians.

1. Introduction

Motivations and Goals

Princeton has experienced many significant changes in the last few decades, including diversifying its student body in terms of gender, race and socioeconomic status, the emergence of a more inclusive process for eating club admittance, and also an academic shift with the rise of computers and digital technology. Throughout this time, many notable events, including numerous protests against the Vietnam War, the increase in drinking age to 21, and visits by famous political world leaders have occurred. Through publishing daily recounts of issues and events, the Daily Princetonian successfully documents Princeton's history. Analyzing the text from articles will document history from a linguistic perspective and allow trends and relationships to be visualized.

The project had two main goals in terms of textual analysis – the first was to track the change in usage of words over time, and the second was to capture the relationships between words at a given point in time. For example, tracking the change in word usage could be finding how many occurrences of the words *girls* and *boys* occurred compared to *women* and *men*, plotting their relative frequencies across different years to show a change in language throughout time. Examples of examining relationships between words include using cosine similarity of word vectors to determine the closest words to *lawnparties* in a given year, determining which word in a group of words doesn't match (e.g. out of the eating clubs Tower, Charter and Colonial, which one is the odd one out?), or even using distance, addition and subtraction of word vectors to determine analogies between certain words, such as the sister college relationship of Mathey to Rocky is the same as the one between Whitman and Forbes.

To accomplish these goals, I used several word embedding models including the bag-of-words model, where the text is represented as a multiset (bag) of words, disregarding word order and grammar, but keeping track of the number of occurrences of each word. This is used to compute the relative frequency of words over time, which in turn can tell us about historical trends. Another model is the word2vec model [7], which uses word embeddings to give each distinct word a unique vector in higher dimensional space, allowing us to quantitatively compare words for semantic similarity and discover word associations and analogies.

Overview of Project

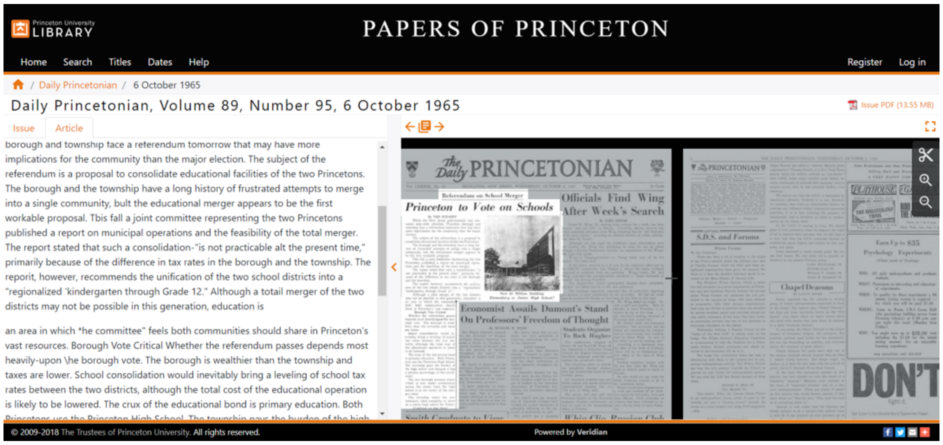
This project explores the Daily Princetonian archives through several natural language processing models. It extends the work of previous researchers and students, allowing a lens through Princeton's history and revealing the complex intricacies of relationships between various words at Princeton.

The Daily Princetonian[1] is the independent student newspaper of Princeton University. The paper began in September 1876 and originally published fortnightly during the academic year until April 1883, weekly until April 1885 and bi-daily from April 1885 to 1892. Published daily since April 1892, it is the second oldest running publication of its kind in the United States and has over 8000 readers. Staffed by Princeton University students, the paper covers both campus news as well as national headlines from a Princeton University perspective. Therefore, the paper provides a unique viewpoint of the history of both Princeton and society from 1876 onwards. From 22 September 1876 to 18 December 2015, a total of 21857 issues have been published, containing a massive amount of not only textual data, but also images and advertisements.

In May 2012, the Princeton University Archives along with the Princeton University Library Digital Initiatives, digitized all the past Daily Princetonian newspapers since its inception in 1876. The newspaper archives were loaded onto the Larry DuPraz Digital Archives [8], a website allowing users to search for keywords. To create the archive, entire copies of newspapers, including earlier ones on microfilm, were sent to Ottawa, Canada, where TIFF images were generated through cameras. The files were then sent to Cambodia via hard drive where the structure of each page was analyzed, and then Optical Character Recognition (OCR) was used to extract the text along with its metadata, including the position of the images. This was finally brought back to the United States where the data was loaded into Veridian, a content-management system which supports

[1] http://www.dailyprincetonian.com/

searching, article extraction, and online browsing. Along with the Daily Princetonian, other Princeton newspapers, including the Princeton University Weekly Bulletin and the Nassau Literary Review have also been digitized and uploaded. The database is online at Princeton University Library's website, where viewers may access, search and browse the publications for personal, scholarly, research or professional use.[2] Prior to this $275,000 digitization, scholars had to sift through hardbound volumes of the 'Prince' in the basement of Mudd Library or tediously scroll through microfilm, limiting the paper's usefulness as a resource for large research projects.


Princeton University LIBRARY
PAPERS OF PRINCETON
Home
Search
Titles
Dates
Help
Register
Log in
Daily Princetonian / 6 October 1965
Issue PDF (13.55 MB)
Issue
Article
The Daily PRINCETONIAN
Princeton to Vote on Schools
Officials Find Wing After Week's Search
Economist Assails Dumont's Stand On Professors' Freedom of Thought
S.D.S. and Forums
Chapel Deacons
PLAYHOUSE
DON'T
© 2009-2018 The Trustees of Princeton University. All rights reserved.
Powered by Veridian

Daily Princetonian, Volume 89, Number 95, 6 October 1965

borough and township face a referendum tomorrow that may have more implications for the community than the major election. The subject of the referendum is a proposal to consolidate educational facilities of the two Princetons. The borough and the township have a long history of frustrated attempts to merge into a single community, bult the educational merger appears to be the first workable proposal. Tbis fall a joint committee representing the two Princetons published a report on municipal operations and the feasibility of the total merger. The report stated that such a consolidation-"is not practicable alt the present time," primarily because of the difference in tax rates in the borough and the township. The reporit, however, recommends the unification of the two school districts into a "regionalized 'kindergarten through Grade 12." Although a totail merger of the two districts may not be possible in this generation, education is

an area in which *he committee" feels both communities should share in Princeton's vast resources. Borough Vote Critical Whether the referendum passes depends most heavily-upon \he borough vote. The borough is wealthier than the township and taxes are lower. School consolidation would inevitably bring a leveling of school tax rates between the two districts, although the total cost of the educational operation is likely to be lowered. The crux of the educational bond is primary education. Both

Figure 1: The Daily Princetonian Archives.

Although this format makes it very easy to find specific articles or issues from a certain date, this doesn't allow researchers to be able to look for trends throughout the data or perform analysis on a large scale with articles spanning many decades. We will scrape all the text from within the articles from all the issues over the last seven decades, and group them by year. Along with some preprocessing, this will allow us to gather the data in a form sufficient to perform natural language analysis and to use data visualization techniques. The results will demonstrate how the use of gendered nouns has changed over the years, as well as academic, social and political shifts. Finally, we will see the relationships between different words from a unique Princeton perspective.

[2] https://rbsc.princeton.edu/databases/papers-princeton

2. Problem Background and Related Work

Since the Daily Princetonian archives were released online in 2012, there have only been two projects that have attempted to analyze the text within the newspaper archives. David M. Liu's work [6] focused on sentence-level sentiment analysis and n-gram visualization, but only on newspaper headlines, completely ignoring the text within the articles. The result was that sentiment analysis was not effective for visualizing trends in the archive, and could not determine bias among articles.

Evelyn Karis's work [5] focused on an intertextual analysis between the Daily Princetonian and the local town newspaper, Town Topics. Her work focused on n-grams to visualize and compare the differing frequencies of words found between the two newspapers. Her work extended Liu's work by focusing on the actual articles themselves rather than headlines, and comparing them to Google's n-gram counter gathered from the raw frequencies of words from Google Books. We extend Karis's work by creating a tool that plots the frequencies of multiple different words on the same chart, allowing comparative analysis between words, instead of comparing the same word across different newspapers.

This project builds upon the findings of the previous two authors by focusing on word embeddings to capture relationships between actual words themselves. One significant difference is that we do not focus on n-grams throughout my work, and we do not compare word frequencies with those found in the Google Books database. The reason is as follows – newspapers are published often daily, and focused primarily on events. By contrast, books are typically published on a much larger time scale, so individual daily events are less likely to be captured in books compared to newspapers. Books are also much more selective about what to publish, so the frequencies of certain words will be distorted, since some events may be covered much more in printed literature compared to in newspapers, which publish consistently on a daily basis.

Due to various problems with sentiment analysis encountered by previous researchers, this project will instead use word embeddings to visualize relationships between words, and use the relative frequencies of words across time to capture shifts in word choice. Instead of comparing between other newspapers and Google Books, we will only compare words to other words within the Daily Princetonian, since we are looking at the shift in words from a Princeton perspective, with the word vectors existing in the context of the Princeton newspaper.

In recent years, many newspapers have switched to digital format, but few prior studies have visualized newspaper archives, or performed natural language analysis on its text. Digitizing newspaper archives has become increasingly common, with many major newspapers such as the New York Times releasing its archives. However, most of these are in either PDF or image formats, and few newspapers have utilized optical character recognition (OCR) technology to transform the articles to text. The Daily Princetonian, published daily, has recorded and documented history for over 100 years, from a unique perspective observed at Princeton. Through digitizing copies of newspapers and publishing the website online, the Princeton University Library has allowed researchers to computationally study this data. This project examines the data and uses textual analysis to examine changes in Princeton over time. Through using the word2vec model, we will be able to identify word associations and semantic similarities, as well as analogies between words.

3. Approach

The most difficult part of the project was gathering the data, and a significant portion of the time was spent on pre-processing the archival data to a format which could be easily used to perform textual analysis. Although all the data was available on the Daily Princetonian Archive site, it was not in a format that was directly accessible or able to be worked with. While it is very easy to find individual articles through searching for titles and dates, it is difficult to gather all the articles from an issue. For example, the words in the article are stored as metadata themselves and there is no easy way to directly extract all the text from the archives.

Another issue was the enormous volume of data available to analyze. With articles dating back from 1876, this proved quite a formidable task. With no consistency checks between various issues of the Daily Princetonian, and the newspaper changing format several times since its inception, a concern was how to make results consistent and how to narrow the scope of this project, allowing more focused findings and deeper analysis to be made.

A decision was made to focus on issues after 1946 for several reasons. Firstly, the Prince did not publish in the three years between 1942 and 1945, instead publishing as The Princeton Bulletin, a Wartime Successor to The Daily Princetonian. This changed the format of the newspaper dramatically, making it rather difficult to compare articles during this period with articles from outside this period. Newspapers after 1946 have stabilized in format, with a steady number of around 150 issues published each year, allowing us to receive consistent results without the concern of different newspaper formats.

In addition, the demographics of Princeton changed significantly after World War II. Many veterans returned to campus, and students whose studies were interrupted by the war also came to campus. This resulted in a massive increase in the undergraduate population and as a result, many aspects of Princeton changed, including a need for a greater number of dormitories and resources including staff. These sudden changes make 1946 a perfect year to start when performing textual analysis on the articles. Furthermore, most of the interesting events related to coeducation, protests and racial issues occurred after 1946 and the events which happened prior to 1945 are far less in frequency compared to events after 1946. The rate of change is much more after 1946, allowing trends to be visualized more easily and relationships to easily be determined.

Lastly, due to differing fonts, the newspapers before 1945, especially ones around the early 20th century, make textual analysis significantly harder, since the same OCR system was used to extract text from all the Daily Princetonian archives regardless of year. This causes many problems for our textual analysis, especially when computing relative frequencies of words, since words that are misspelled will not add to our frequency tables, and the same words will tend to be consistently misspelled with older fonts.

4. Implementation

The project was implemented in several steps. Firstly, the available data was collected, cleaned and organized into a format that could easily be parsed to create natural language processing models.

Next, the words themselves from the articles were tokenized to compute the relative frequency of words. Lastly, a Word2Vec model [4] was used to capture the relationships between different words and data visualization tools applied to display various political, cultural and social trends through language. Each step involved individual challenges, which will be discussed below.

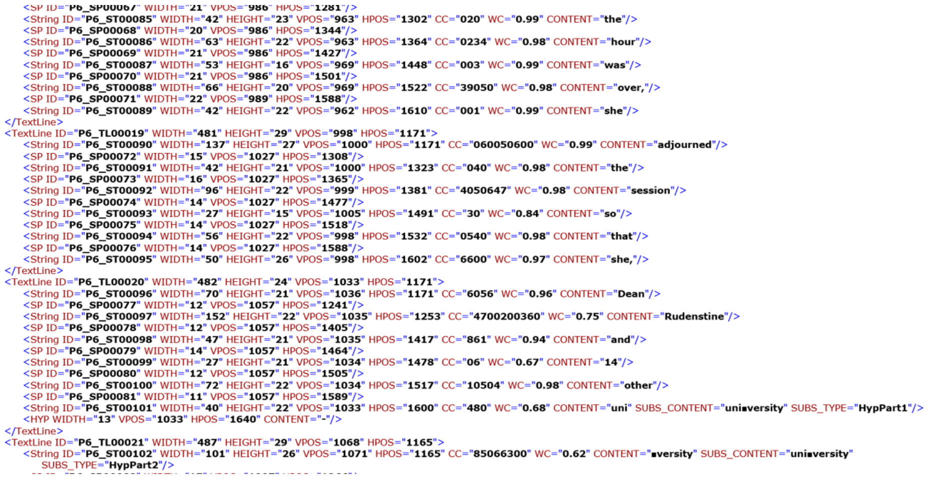
Scraping

The newspaper archives were hosted on a platform called Veridian[3], which supports newspaper digitization projects, providing online search and display of digitized newspaper collections. The Daily Princetonian archives use the METS/ALTO [2] markup standard, the same as the Library of Congress's Newspaper Digitization Project, allowing the resource to be sustained as software changes over time.

METS and ALTO are XML standards maintained by the Library of Congress. The Metadata Encoding and Transmission Standard (METS) is a schema describing the structure of a complex object, for example, a digitized newspaper issue, but not the actual textual content of the object itself, which is represented in the Analyzed Layout and Text Object (ALTO) standard. As well as encoding all the digitized text, ALTO also captures the spatial coordinates of every column, line and word on the page including styles and layouts. As seen in Figure 2, we can extract the text from the article from CONTENT, but the XML also stores the text's width, height and horizontal/vertical positions. At the bottom of the figure, we can even see that it recognizes that the word *university* has been split by a hyphen into *uni-versity*.

The combination of METS and ALTO is the current industry standard for newspaper digitization and works very well for the Princeton University Library's goal to digitize all the newspaper archives and display them in a format accessible to visitors and to search for keywords, but this makes it quite difficult for individuals who want to work directly with the text from the articles instead of searching for a specific issue. Currently, the website supports browsing by title or date and searching by keywords, which makes it very easy for people who know what they are searching

[3] https://www.veridiansoftware.com/knowledge-base/veridian/



```
    <SP ID="P6_SP00067" WIDTH="21" VPOS="986" HPOS="1281"/>
    <String ID="P6_ST00085" WIDTH="42" HEIGHT="23" VPOS="963" HPOS="1302" CC="020" WC="0.99" CONTENT="the"/>
    <SP ID="P6_SP00068" WIDTH="20" VPOS="986" HPOS="1344"/>
    <String ID="P6_ST00086" WIDTH="63" HEIGHT="22" VPOS="963" HPOS="1364" CC="0234" WC="0.98" CONTENT="hour"/>
    <SP ID="P6_SP00069" WIDTH="21" VPOS="986" HPOS="1427"/>
    <String ID="P6_ST00087" WIDTH="53" HEIGHT="16" VPOS="969" HPOS="1448" CC="003" WC="0.99" CONTENT="was"/>
    <SP ID="P6_SP00070" WIDTH="21" VPOS="986" HPOS="1501"/>
    <String ID="P6_ST00088" WIDTH="66" HEIGHT="20" VPOS="969" HPOS="1522" CC="39050" WC="0.98" CONTENT="over,"/>
    <SP ID="P6_SP00071" WIDTH="22" VPOS="989" HPOS="1588"/>
    <String ID="P6_ST00089" WIDTH="42" HEIGHT="22" VPOS="962" HPOS="1610" CC="001" WC="0.99" CONTENT="she"/>
</TextLine>
<TextLine ID="P6_TL00019" WIDTH="481" HEIGHT="29" VPOS="998" HPOS="1171">
    <String ID="P6_ST00090" WIDTH="137" HEIGHT="27" VPOS="1000" HPOS="1171" CC="060050600" WC="0.99" CONTENT="adjourned"/>
    <SP ID="P6_SP00072" WIDTH="15" VPOS="1027" HPOS="1308"/>
    <String ID="P6_ST00091" WIDTH="42" HEIGHT="21" VPOS="1000" HPOS="1323" CC="040" WC="0.98" CONTENT="the"/>
    <SP ID="P6_SP00073" WIDTH="16" VPOS="1027" HPOS="1365"/>
    <String ID="P6_ST00092" WIDTH="96" HEIGHT="22" VPOS="999" HPOS="1381" CC="4050647" WC="0.98" CONTENT="session"/>
    <SP ID="P6_SP00074" WIDTH="14" VPOS="1027" HPOS="1477"/>
    <String ID="P6_ST00093" WIDTH="27" HEIGHT="15" VPOS="1005" HPOS="1491" CC="30" WC="0.84" CONTENT="so"/>
    <SP ID="P6_SP00075" WIDTH="14" VPOS="1027" HPOS="1518"/>
    <String ID="P6_ST00094" WIDTH="56" HEIGHT="22" VPOS="998" HPOS="1532" CC="0540" WC="0.98" CONTENT="that"/>
    <SP ID="P6_SP00076" WIDTH="14" VPOS="1027" HPOS="1588"/>
    <String ID="P6_ST00095" WIDTH="50" HEIGHT="26" VPOS="998" HPOS="1602" CC="6600" WC="0.97" CONTENT="she,"/>
</TextLine>
<TextLine ID="P6_TL00020" WIDTH="482" HEIGHT="24" VPOS="1033" HPOS="1171">
    <String ID="P6_ST00096" WIDTH="70" HEIGHT="21" VPOS="1036" HPOS="1171" CC="6056" WC="0.96" CONTENT="Dean"/>
    <SP ID="P6_SP00077" WIDTH="12" VPOS="1057" HPOS="1241"/>
    <String ID="P6_ST00097" WIDTH="152" HEIGHT="22" VPOS="1035" HPOS="1253" CC="4700200360" WC="0.75" CONTENT="Rudenstine"/>
    <SP ID="P6_SP00078" WIDTH="12" VPOS="1057" HPOS="1405"/>
    <String ID="P6_ST00098" WIDTH="47" HEIGHT="21" VPOS="1035" HPOS="1417" CC="861" WC="0.94" CONTENT="and"/>
    <SP ID="P6_SP00079" WIDTH="14" VPOS="1057" HPOS="1464"/>
    <String ID="P6_ST00099" WIDTH="27" HEIGHT="21" VPOS="1034" HPOS="1478" CC="06" WC="0.67" CONTENT="14"/>
    <SP ID="P6_SP00080" WIDTH="12" VPOS="1057" HPOS="1505"/>
    <String ID="P6_ST00100" WIDTH="72" HEIGHT="22" VPOS="1034" HPOS="1517" CC="10504" WC="0.98" CONTENT="other"/>
    <SP ID="P6_SP00081" WIDTH="11" VPOS="1057" HPOS="1589"/>
    <String ID="P6_ST00101" WIDTH="40" HEIGHT="22" VPOS="1033" HPOS="1600" CC="480" WC="0.68" CONTENT="uni" SUBS_CONTENT="uni▪versity" SUBS_TYPE="HypPart1"/>
    <HYP WIDTH="13" VPOS="1033" HPOS="1640" CONTENT="-"/>
</TextLine>
<TextLine ID="P6_TL00021" WIDTH="487" HEIGHT="29" VPOS="1068" HPOS="1165">
    <String ID="P6_ST00102" WIDTH="101" HEIGHT="26" VPOS="1071" HPOS="1165" CC="85066300" WC="0.62" CONTENT="▪versity" SUBS_CONTENT="uni▪versity"
      SUBS_TYPE="HypPart2"/>
```

Figure 2: The XML files contain both the word and metadata regarding its position on the page.

for to find their information. However, it makes textual analysis rather difficult, because it makes it almost impossible to discover trends in language.

Fortunately, Veridian's XML API allows users to extract text and images out of Veridian in a structured form. Using document identifiers along with the hierarchical position of the logical section (article) within the document, an XML response containing just the data for that logical section in a structured form can be obtained. For example, the URL http://theprince.princeton.edu/princetonperiodicals/?a=d&d=Princetonian20111110-01.2.9&f=XML can be used to extract the article with ID 9 within the Daily Princetonian issue with ID *Princetonian20111110-01* (the .2 following the issue ID precedes the article ID). This could be parsed to look for the LogicalSectionTextHTML element, which contains the text of the article, including its headline in the first HTML <p> tag.

The entire process would be as follows. First, make an HTTP request to Veridian to query what issues of the Princetonian we would like, perhaps within a certain year/time range. For each of those issues, we would gather its articles, also known as "logical sections", and make HTTP requests to get their content as XML. We would then parse the XML response received and extract the article along with the headline, which we would aggregate into a structured document for further analysis.

In terms of the tools and technologies used for the project, the whole analysis was performed using Python for several reasons. Firstly, Python supports Natural Language Analysis with NLTK

[1] (Natural Language Toolkit), which is a Python library for NLP. It supports word tokenization, stemming, part-of-speech tagging, collocations and even named-entity recognition. All the analysis was performed inside a Jupyter Notebook, which allows interactive computation as well as support for inline data visualization. To create the word embedding model, we used Gensim [3], which is an open-source Python implementation of Word2Vec, using NumPy and C for performance. Finally, the plots were all generated using Matplotlib and Seaborn which work perfectly with Jupyter Notebook for data visualization.

Processing the data

After we gathered all the text from the articles, we processed the data to make it easier to work with. Firstly, we grouped the text by year, since in order to track changes over time, we must have an interval to measure over, and a year was the best choice, given we have exactly 70 intervals, a number that is neither too large or small. Grouping the articles instead by academic year (September – May) was briefly considered, but ruled out given that grouping the articles would be considerably more difficult, and also the differences in results would be insignificant.

First, we converted all the words to lower case, so that words like ‘Bonfire’ and ‘bonfire’ will not be treated as two different words. We then remove stop words from the analysis, so that words like ‘a’, and ‘the’ will not impact our results, otherwise, if we look for words that occur in a similar context to Princeton, we may find several prepositions, since they occur in close proximity to many other words. Next, we use NLTK’s built-in tokenizer, which separates the words into a list. An advantage of NLTK is that it recognizes contractions and “didn’t” is split into two words, “did” and “not”. The list of tokens is then converted into an NLTK Text object. Due to irregular spacing in some earlier editions of the newspaper, some words such as *cosmopolitan* and *costume* were incorrectly split by OCR into *cos mopolitan* and *cos tume*, but fortunately occurrences like these were low enough not to affect our results at all.

To compute the frequency of a word for a given year, we wrote a function which takes in a string and an NLTK Text object, and returns the number of times the string occurs within the text, divided by the total number of words to calculate the relative frequency. This is then repeated for each of the years from 1946 to 2015, obtaining a list of 70 frequencies which are then plotted using Matplotlib under a line graph. We also wrote functions to compute and plot multiple frequencies of several words together in one graph to show the correlations between two different words over time. Since the words from each year of the newspaper were already tokenized and cast into an NLTK Text object, this computing the frequencies was a very efficient procedure since we can store all the words along with their frequencies as key-value pairs. As seen in Figure 3, the total number of words in each year of the Daily Princetonian ranges between 1.27 million in 1946 to 2.72 million in 2006, which is consistent with the number of words we expect to find in each issue, and provides us with a large enough sample to ensure we will receive statistically significant results.

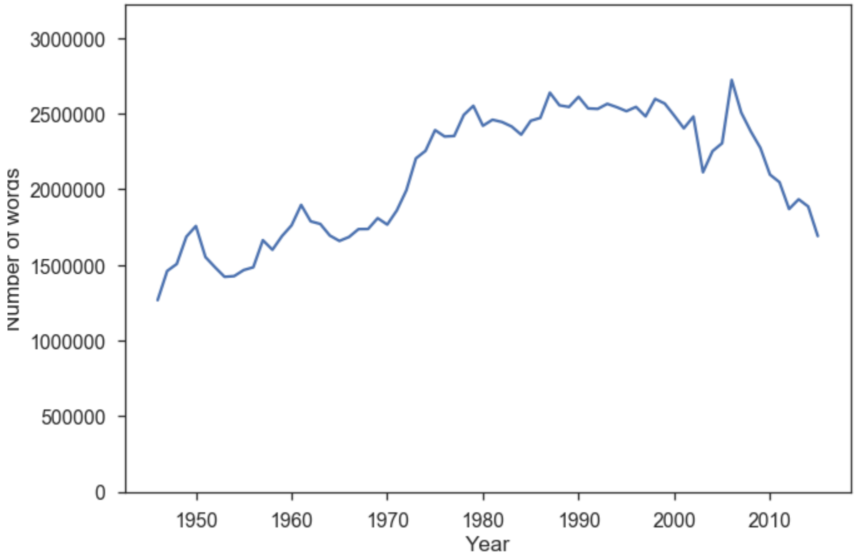


Figure 3: Number of words in each year of the Daily Princetonian.

Word2Vec

Word2vec [7] is a model used to develop word embeddings, which are vector representations of words. They are capable of capturing the context of a word within a document and also its similarity and relationships with other words. It was developed at Google in 2013 by a team of researchers led by Tomas Mikolov and uses a shallow, two-layer neural network.

There are two methods used in word2vec to construct such an embedding: the skip gram model and the continuous bag of words (CBOW) model. Both involve neural networks and are illustrated in Figure 4. With CBOW, the model predicts the current word from a window of surrounding words, however, with skip grams, the model uses the current word to predict the surrounding window of context words, with closer words weighted more heavily. Both have their own advantages and disadvantages: skip grams works well for rare words but CBOW is much faster and has better representations for more frequent words. Upon experimentation, it was found that both models had very similar results for the Daily Princetonian text dataset, and a decision was made to stick with the default choice of CBOW given its slightly faster implementation and the fact that our queries were all common words in the context of Princeton.

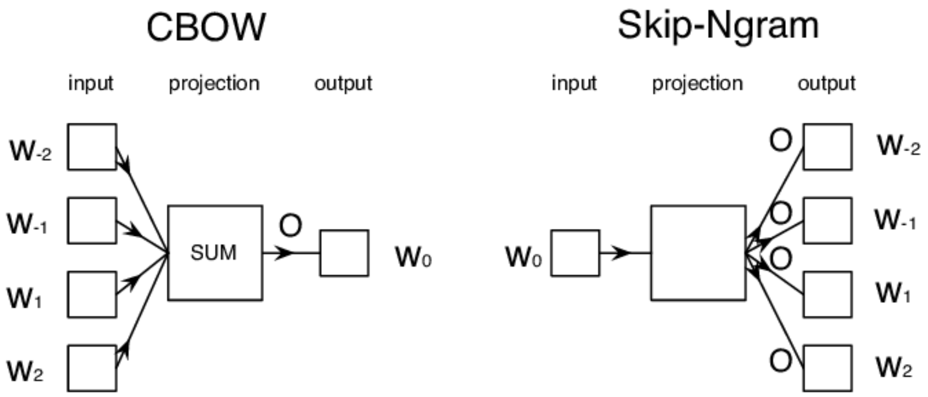
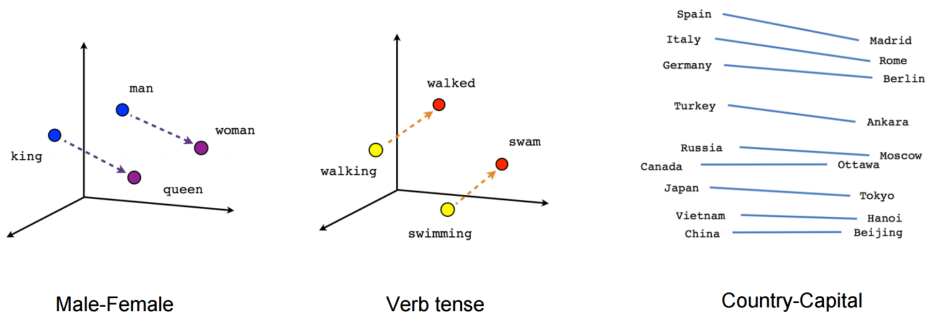


Figure 4: Comparison between the CBOW and skip-gram model for word2vec.

One disadvantage of using word2vec is that the results of word2vec training are highly sensitive to the initial parameters [9]. The word2vec model is quite complex and has many parameters including the dimensionality of vectors, number of epochs and learning rate for training, the sliding window size (how many words before and after a given word should be used), whether or not to remove words with low frequency and whether to train with hierarchical softmax and/or negative sampling. Using multiple worker threads also makes the generated vectors non-deterministic due to randomness from OS thread scheduling. For a fully deterministically-reproducible result, we would also need to set the PYTHONHASHSEED environment variable. Given that the randomness caused a negligible difference in results, and the same conclusions were able to be observed through repeated runs, a decision was made to allow the vectors to slightly vary when a new model is trained, since the relationships between the vectors all stay the same.

The time required to train the model was not a concern, since after training the model once, we obtain the vector space with all the words and their corresponding vectors, meaning that actions such as computing cosine similarity or finding distance between two vectors are instant in that we don't have to recompute vectors. After the vector space has been created, all the word vectors are stored, making computation such as similarity and distance as simple as retrieving the two coordinates from a key-value dictionary, eliminating the need to re-create the vectors. We can use simple vector addition and subtraction to find analogies between different vectors, such as that the relationship between King to Queen is the same as that between man to woman, as seen in Figure 5.



Male-Female

Verb tense

Country-Capital

Figure 5: Vector addition and subtraction can be used to visualize word analogies.

Since we use word2vec to model the relationships between actual words in one text, and NLTK tokens to model the change in frequency of one word across many different texts, the approaches are slightly different. For example, for the NLTK Text object, only a bag of words model was needed. This could be a Python Counter object, which is a dictionary of key-value pairs of the word and its frequency. However, for word2vec the location of each word in relation to its neighboring words is very important. For this reason, we don't need to use a stemmer to remove suffixes (we will later see with word tense) and can largely leave the original text untouched. Words that are incorrectly captured by OCR affect the model differently too, instead of disturbing the frequency distribution as in the NLTK model, with word2vec, misspellings of the same word will frequently have almost identical vector representations, since they occur in the same context, and we will inadvertently have that they are very similar to each other.

An important issue to focus on is the problem of underfitting and overfitting. If our vector dimensions are too small, then we will have too many words all similar to each other, since the low dimensions cause greater similarity, leading to underfitting. However, the opposite occurs if we have too many dimensions, making our model sparse and all the vectors far apart from each other. As an extreme example, if we have more dimensions than the number of words, then each word vector will have principal components in their own dimension, making similarity between all words zero. Another important parameter was the window length, which is equal to the length of surrounding words that we use to generate a vector for each word. If this is too large, then we will have overfitting since words that are far apart from each other in a sentence will appear as 'similar', and a small window size will lead to underfitting due to not many similar words.

Visualizing data with word2vec was more challenging, given that the vectors themselves were often 200-dimensional. Two techniques to plot higher dimensional data are PCA (Principal Components Analysis) and t-SNE (T-Distributed Stochastic Neighboring Entities) which both reduce the dimension of the data into a lower number. PCA reduces the number of dimensions on a dataset while retaining as much information as possible by using the eigenvectors of the covariance matrix associated to the largest eigenvalues to capture the directions of maximum variance in the dataset. T-SNE is another dimensionality reduction technique which models each vector as a two-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are distant points with high probability. We tried plotting vectors using both techniques, but although the two dimensions definitely hold some information, it is not enough to clearly display the relationships between most vectors, especially since so much information has been lost through the reduction and the two axes are now meaningless. As seen in Figure 6, t-SNE can be used to visualize the vectors in two dimensions, however, the two axes do not have any real meaning, and distance/similarity between vectors is somewhat distorted. In addition, due to the probabilistic nature of t-SNE, the result is not deterministic and slightly different results can be generated with the same set of vectors.

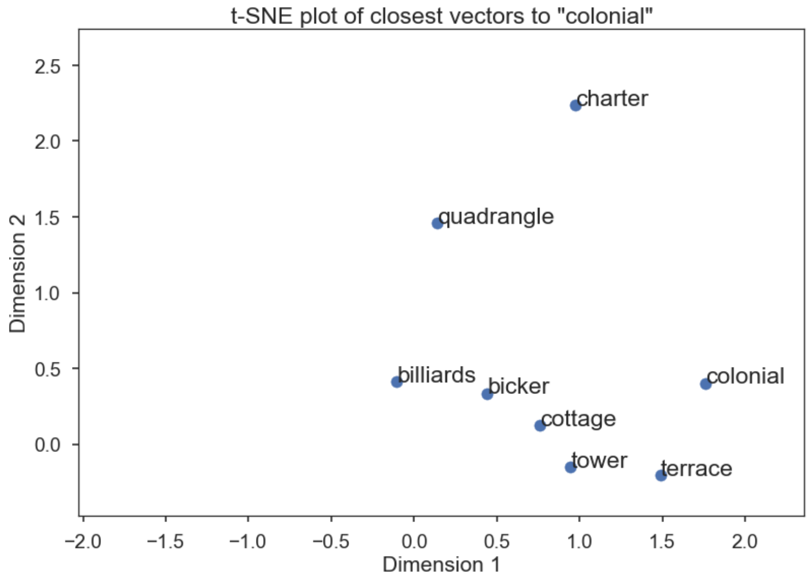


Figure 6: A t-SNE visualization of the vectors closest to the word ‘colonial’.

To start off, we used the default parameters (100 dimensions, window of 5 words between current and predicted, CBOW training) to generate vectors for words in the year 2010. Early results looked promising, but contained many misspellings. For example, the closest word by a large margin to *Kernighan* was the misspelling *Kemighan*, which makes sense since both words would occur in the same context, meaning semantic similarity should be one. Similarly, the word *Perm* appeared frequently where *Penn* was expected, and the word *ton* even appeared for *Princeton*. A noteworthy observation was that misspelled words were grouped with high similarity to other misspelled words. While this may seem intriguing, there is a simple explanation. The text was generated using OCR on image files made by photographing physical newspapers. In articles where the OCR doesn’t correctly predict the word, there is a good chance that words in close proximity to it are also misspelled.

We measured similarity between two word vectors by their cosine distance, which is equal to the cosine of the angle between the two vectors, given by their dot product divided by the product of their magnitudes. This distance is always between -1 and 1 by the Cauchy-Schwarz inequality, with vectors that are semantically similar having distance closer to 1. This can be used to determine which word doesn't belong in a group of words. For example, it can tell us that out of the three Eating Clubs, Colonial, Charter and Tower, that Tower is the odd one out. This would make sense, given that Tower is a bicker club while the other two are both sign-in. Upon searching for vectors closest to *Harvard* (see Table 1), the word2vec model says that Harvard is closest to Yale, with cosine similarity 0.845, followed by Dartmouth, Cornell, Brown, and Columbia, which are all Ivy League Colleges. The word *Princeton* did not appear as one of the most similar words likely because the Daily Princetonian is based in Princeton, meaning that the word *Princeton* frequently occurs in contexts outside of Ivy League Colleges, leaving it with a completely different vector representation compared to other universities.

Word	**Cosine Similarity**
Yale	0.7954
Cornell	0.7588
Dartmouth	0.7565
Brown	0.7437
Columbia	0.6943

Table 1: Words most similar to Harvard.

5. Results and Evaluation

Relative frequencies of words

After all the text from the articles are gathered into a text file for each corresponding year, we read it into a variable in Python, stripping the words into lowercase in the process to make sure words like *Professor* and *professor* are grouped together. The text is then tokenized and transformed into an NLTK Text object. To compute the relative frequency, we divide the total number of occurrences by the total number of words in each year, and then plot the data using Matplotlib. We use Seaborn to style the graph. On the vertical column, the frequency is in percentages, so a frequency of 0.02 means that there are two occurrences of that word for every 10000 words. However, we are not concerned with this at all, since we are looking at the change in relative frequency of the words, making the scaling irrelevant. Due to the large number of results, most of the plots have been moved to the appendix.

As an example, let's see occurrences of the word *olympics* throughout the Daily Princetonian (our data is now case-insensitive after preprocessing). We can see in Figure 7 that initially, the word is roughly periodic, with a peak during years that are multiples of four, due to the Summer Olympics and troughs of close to zero frequency on other years. This would make sense, given that many athletes would be either current or graduated Princeton students. The spike in 1980 corresponds to the Nude Olympics becoming a well-established tradition by 1980, making the word *olympics* a much more popular word, until its demise in 1999 when the University banned Nude Olympics. Something worthwhile to mention is that in 1994, the International Olympic Committee decided to hold the Winter Olympic Games two years apart from the Summer Games rather than during the same years. This now corresponds to peaks every even year, with troughs on odd years, which is consistent with post-1994 data.

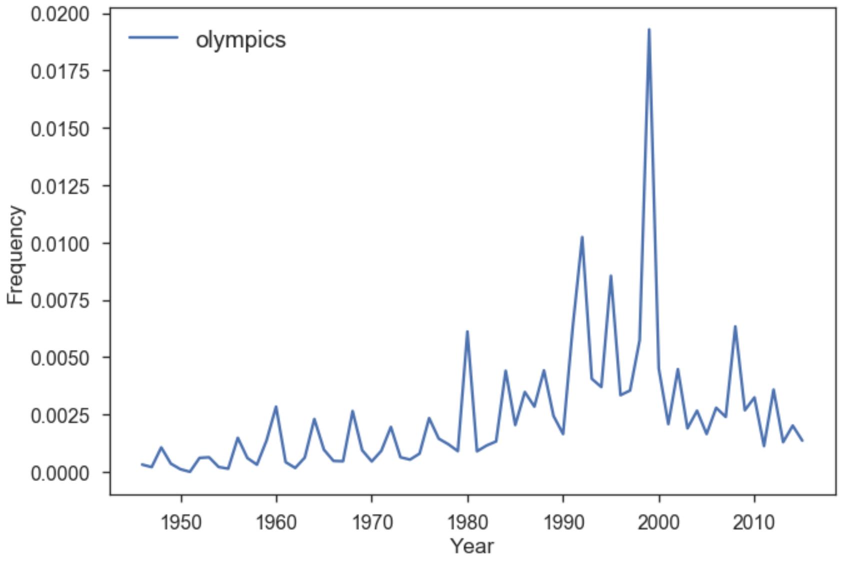


Figure 7: Relative frequency of the word 'Olympics'.

Comparing the words 'Oriental' and 'Asian' in Figure 8, we can see that before the mid-1950s, there was no mention of Asian, and Oriental was used extensively. However, around 1970, usage of

Oriental dropped sharply and the use of Asian steadily increased, with Oriental virtually unused since the late 1980s. An interesting extension would be to plot the proportion of the word usage ‘Oriental’ vs ‘Asian’ over time, rather than comparing their relative frequencies.

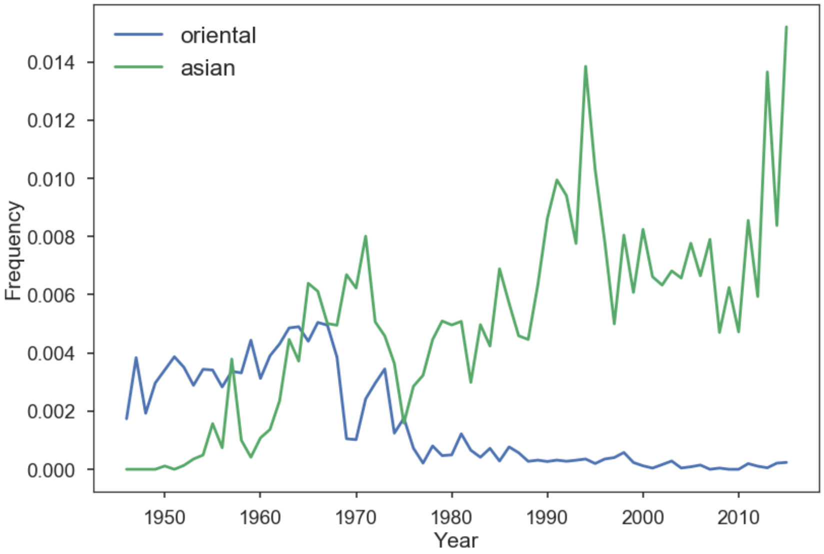


Figure 8: Relative frequency of the word ‘Oriental’ vs ‘Asian’.

Comparing the number of occurrences of the word ‘computer’ to that of various sciences in Figure 9, we can see that the term ‘computer’ has become dominant after the mid-1960s, emerging from almost nothing in 1950, while the sciences have remained relatively stable. Note that ‘mathematics’ had a sharp decrease in the year where ‘computer’ emerged from almost having no previous mentions.

Comparing the different spellings of theater and theatre in Figure 10, we can see that the spelling ‘theatre’ actually dominated usage in the Daily Princetonian until a sharp decrease in the mid-1980s. Both terms are used interchangeably now, with even McCarter Theatre and the Garden Theatre frequently using the -er forms.

Comparing the occurrences of ‘girls’ and ‘boys’ in Figure 11, we can see that their usage has gone down, with the words largely having been replaced with ‘women’ and ‘men’ over time as

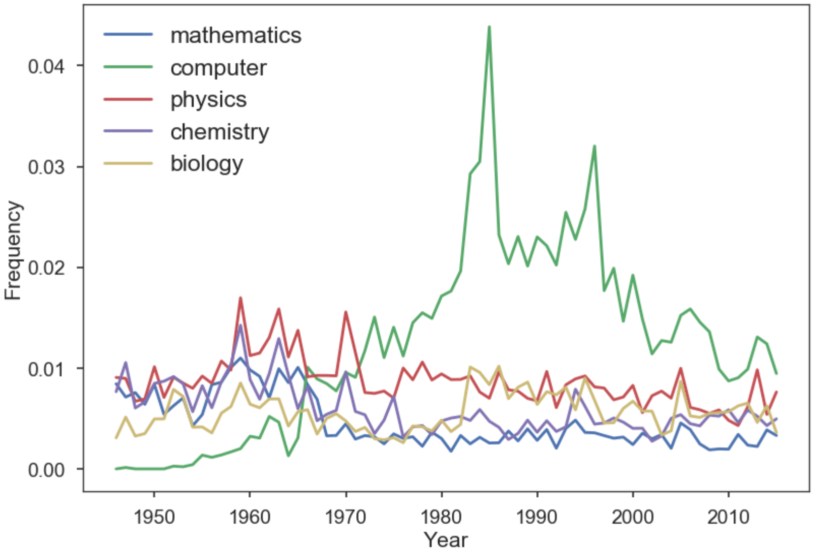


Figure 9: Relative frequency of computer vs various sciences.

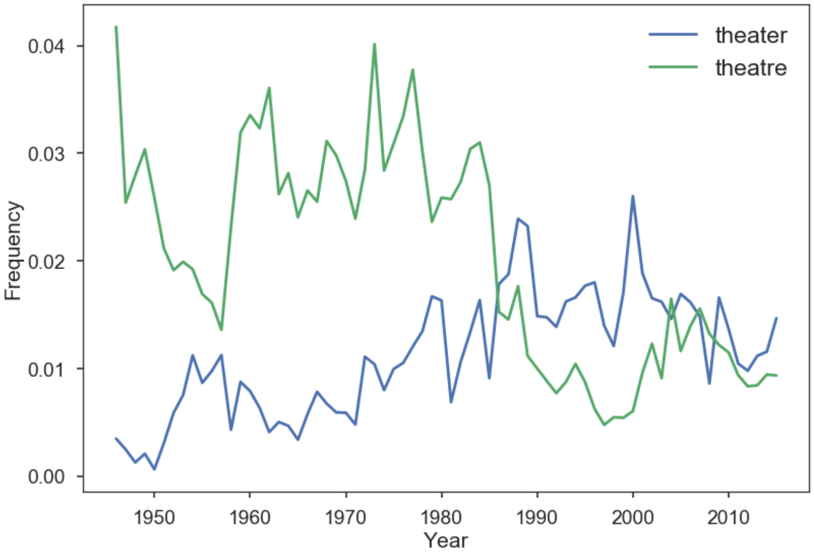


Figure 10: Relative frequency of the word ‘Theater’ vs ‘Theatre’.

seen in Figure 12, but there is a recent resurgence in the use of ‘girls’. The university first admitted women in 1969, as shown by the large spike in relative frequency.

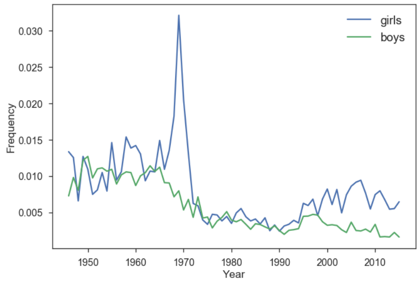


Figure 11: Relative frequency 'girls' vs 'boys'.

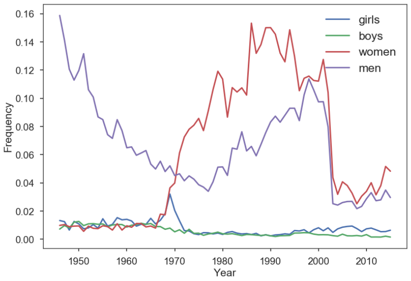


Figure 12: Relative frequency 'women' vs 'men'.

6. Summary

Conclusions

This project used the past seventy years of Daily Princetonian news articles to analyze cultural and historical trends at Princeton through shifts in word usage over time. The articles were scraped from the online database, and tokenized to perform textual analysis. The frequencies of words were plotted using data visualization tools in Python, charting usage over time and relating them to changes in society and attitude. In addition, a word2vec model was used to examine relationships between different words through a Princeton perspective. Through this, we were able to see how closely related words using cosine similarity between vectors, using addition and subtraction of vectors to determine analogies between words and using distances to measure word associations. Some examples of results we found included the different language used to describe gender and race, the relationships between various Eating Clubs and Residential Colleges on campus, academic shifts with the advent of computers and we could even match specific Princeton events such as grade deflation, bonfires and the Nude Olympics. These results, along with the textual analysis and visualization tools, can be used by future scholars and researchers to document historical trends and linguistic shifts over time.

Future Work

This project examines the archives from the Daily Princetonian from 1946 to 2015 and uses several natural language processing models to analyze the text within the articles for linguistic, social, political and academic trends. Due to the limitations as a one-semester independent research project, there are several aspects which would be worth considering for future work. Firstly, the Daily Princetonian has digital archives dating back to 1876, meaning we had another 70 years of data that we could work with. Since the Daily Princetonian was published in a different format, and not always daily, there would need considerable post-processing of the results and data to ensure consistency in style and that our findings are not affected by the change in publication. Another suggestion is to compare the findings at Princeton to those of similar collegiate newspapers, such as the Harvard Crimson and the Yale Daily News, or even newspapers like the New York Times. This would give several interesting findings, such as, did the shift from 'girls' and 'boys' to 'women' and 'men' happen at the same time? Maybe much earlier, later or even not at all? It would be particularly interesting given that the universities allowed coeducation at different times, and each would have had differing campus perspectives during major events such as rallies, protests and legal changes.

Another idea could be to expand on the natural language processing models used to analyze the text. We could utilize sentence2vec, which assigns a vector to each sentence equal to the arithmetic mean of all the word vectors of the words within the sentence. This method works best for computing distance and similarity between tweets, and it would naturally extend to performing very well for semantic similarity between headlines, which can then be used for sentiment analysis, extending Liu's work. This idea could be extended even further to doc2vec, which provides a vector for each document, which can range in any length, from paragraphs to actual articles. This could then be used to measure similarity between articles, and possibly look for bias in reporting, or examine shifts in the publication of articles over time.

More visualization tools, especially for visualizing the vectors themselves, or distance and similarity between vectors could be added, however, this would be quite difficult in two dimensions and contain many limitations. Another suggestion would be to compare similarity between the same words over time, but since we train a different word2vec model for each year of articles, and given that results of word2vec are extremely sensitive to initial parametrization, this would be difficult as the results may not be consistent, and it wouldn't be clear whether differences are statistically significant. Finally, we plan to release the work soon as a user-friendly web interface, where users can search for words, or groups of words, and visualize the results for themselves, extending the Daily Princetonian's digital archives to allow users to directly perform analysis themselves.

7. Acknowledgements

I would firstly like to thank Professor Brian Kernighan for his constant support and guidance throughout the entire semester and providing me with endless ideas on interesting topics to explore. His advice has been invaluable and without his generous time and feedback, this project would not have been possible. I would like to thank the Princeton University Library for producing the archives, in particular University Archivist Daniel Linke from the Mudd Manuscript Library and Clifford Wulfman for introducing me to the archives, providing me with the raw files, teaching me how to extract the article text from those files and giving countless suggestions on historical issues to explore at Princeton. Next, I would like to thank all the people at the Center for Digital Humanities for providing me with potential ideas for projects, particularly Nick Budak, who taught me how to scrape the articles from Veridian's XML API. Furthermore, I would like to thank David Liu and Evelyn Karis, who gave me plenty of recommendations and insight in working with the data. Lastly, I want to thank my family and friends, for all their support through the project.

References

[1] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python — Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009.

[2] S. Boddie, "What is METS/ALTO?" 2014. Available: https://www.veridiansoftware.com/knowledge-base/metsalto/

[3] R. Řehůřek, "gensim: models.word2vec – Word2vec embeddings," 2018. Available: https://radimrehurek.com/gensim/models/word2vec.html

[4] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *CoRR*, vol. abs/1402.3722, 2014. Available: http://arxiv.org/abs/1402.3722

[5] E. Karis, *The Town and Gown: An Intertextual Analysis of Town Topics and the Daily Princetonian*. Princeton University, 2017.

[6] D. M. Liu, *Discovering Princeton's History: A textual analysis of collegiate newspaper headlines*. Princeton University, 2016.

[7] T. Mikolov *et al.*, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. Available: http://arxiv.org/abs/1301.3781

[8] Princeton University Library Digital Initiatives, "The Daily Princetonian: Larry DuPraz Digital Archives." Available: http://theprince.princeton.edu/princetonperiodicals/cgi-bin/princetonperiodicals

[9] X. Rong, "word2vec parameter learning explained," *CoRR*, vol. abs/1411.2738, 2014. Available: http://arxiv.org/abs/1411.2738

I pledge my honor that this paper represents my own work in accordance with University regulations. — Yang Song

Appendix

Unfortunately due to length restrictions, not all the results gathered could be compiled into this report, and regrettably many of them had to be omitted. The appendix here contains a number of findings worth mentioning. As part of intended future work, the code will be released online to create a web application where users can select their own models to perform textual analysis and data visualization on their own topics. This will be done with the Princeton University Library's help to allow data from the Daily Princetonian Archives to be more accessible to researchers, scholars and enthusiasts.

Table 2 shows the words most similar to Colonial. It is worth noting that Charter, Terrace and Quadrangle are all sign-in clubs like Colonial, which makes sense that they are most similar. Eating clubs would be mentioned where bicker is also mentioned, explaining the presence of *bicker*. The word *Barbadian* is seemingly unrelated, but after closer observation, it appears that it is more related to colonization than with the eating club.

Word	**Cosine Similarity**
Charter	0.6530
Terrace	0.5729
Quadrangle	0.5572
Bicker	0.5538
Barbadian	0.5492

Table 2: Words most similar to Colonial.

Table 3 shows the words most similar to the vector given by Mathey - Rocky + Forbes. The result is that Mathey - Rocky + Forbes = Whitman, or Mathey - Rocky = Whitman - Forbes. It is worth noting that Rocky is the two-year sister college of Mathey, and Forbes is the two-year sister college of Whitman. Therefore this shows that our results were successfully able to capture the relationships and analogies between Princeton entities. The result contained many other residential colleges, which makes sense given that the vectors spanned by the various residential colleges should be close together.

Word	Cosine Similarity
Whitman	0.6540
Rockefeller	0.5797
Residential	0.5706
Butler	0.5479

Table 3: Words most similar to Mathey - Rocky + Forbes.

Table 4 shows the words most similar to the vector given by Crimson - Harvard + Yale. The result is that Harvard to Crimson is equivalent to Yale to Bulldogs, representing the relationships between each university to its mascot. The reason why we used Crimson - Harvard and not Tiger - Princeton is because the word Princeton is also associated with other terms rather than Princeton University, and the vector subtraction would not perfectly represent the relationship between a university and its mascot. For example, Princeton could refer to the location of the town itself or even to groups such as the Princeton University Orchestra but Harvard would only refer to the University.

Word	Cosine Similarity
Bulldogs	0.7139
Quakers	0.6901
Bison	0.6731
Colgate	0.6569

Table 4: Words most similar to Crimson - Harvard + Yale.

Similarly, Table 5 shows that the mascot of Penn is Quakers. It is worth noting the closeness of the result between Quakers and Bulldogs, and given slightly different initial parameterization, it is likely that Bulldogs will be more similar to Quakers, showing that the word2vec model does indeed have its limitations.

To test the canonical word2vec example of verb tense, Table 6 shows the example between walking and walking compared to running and ran. Ran narrowly came ahead of injured for word similarity, ahead of starting and Culbreath, who is a former running back for the Princeton Tigers Football Team who came into the news in 2010 (the year for which we developed this particular

Word	Cosine Similarity
Quakers	0.6963
Bulldogs	0.6884
Hawks	0.6853
Bears	0.6440

Table 5: Words most similar to Crimson - Harvard + Penn.

model) after he was diagnosed with the rare disease aplastic anemia. The closeness in similarity is helped by the fact that running is also a common word in sports, especially football.

Word	Cosine Similarity
Ran	0.5300
Injured	0.5291
Starting	0.5071
Culbreath	0.5039

Table 6: Words most similar to Walked - Walking + Running.

Table 7 shows that Chinese - China + Korea = Korean by a large margin, with their capitals Beijing and Seoul also appearing as similar words. It is not immediately clear how the work Turk came into close association with either of the words. Further examination could be easily performed through searching the Daily Princetonian for the work Turk in association with any of the other words.

Word	Cosine Similarity
Korean	0.6704
Turk	0.5705
Beijing	0.5334
Seoul	0.5270

Table 7: Words most similar to Chinese - China + Korea.

Table 8 shows the words with the most semantic similarity to the word Negro in 1970. Along with the following, words such as Chicago, Daley and Cairo were also found to be quite similar. In 2010, the word Negro was not found in our vector space vocabulary, meaning that it occurred less than 5 times in total throughout articles in 2010. This illustrates an example of changing word usage. An interesting extension of this project would be to capture and visualize the change in relationships between words across time using word2vec.

Word	Cosine Similarity
Killed	0.6254
Wounded	0.6191
Born	0.6186
Riots	0.5995

Table 8: Words most similar to Negro in 1970.

Table 9 shows the words with the most semantic similarity to the word Communist in 1970. Other similar words included Cambodia, Troops, Cong and Wars. Thieu was the President of South Vietnam during this period. Words similar to Vietnam in 1970 include Cambodia, Indochina, Laos, Invasion and War. The 1970s was during the height of Vietnam War protests at Princeton University.

Word	Cosine Similarity
Viet	0.7291
Nam	0.7185
Bombing	0.7047
Laos	0.7033
Thieu	0.6934

Table 9: Words most similar to Communist in 1970.

Comparing the number of occurrences of the names of various technology companies in Figure 13, we can see that none of them have any mentions before 1980, with all mentions of Amazon referring to either the river or rainforest. Microsoft quickly rose, later to be replaced by Google and then Facebook, which now has more mentions than the other three combined.

Comparing the number of occurrences of the word ‘alcohol’ in Figure 14, we can see there was a large spike when alcohol laws were changed in New Jersey to raise the legal drinking age from 18 to 21. Given that this is the age of most college undergraduates, the usage of this word has increased significantly as a result.

Comparing the number of occurrences of the word ‘Einstein’ in Figure 15, we can see there was a large spike during the year of Einstein’s death, but also the 100th anniversary of his birth, something which we confirm when we search for the word in the newspaper during those years.

Looking back to Figure 16, we have now added the words ‘sororities’ and ‘sorority’. It seems that over the recent years, the emergence of sororities has caused an increase in the number of

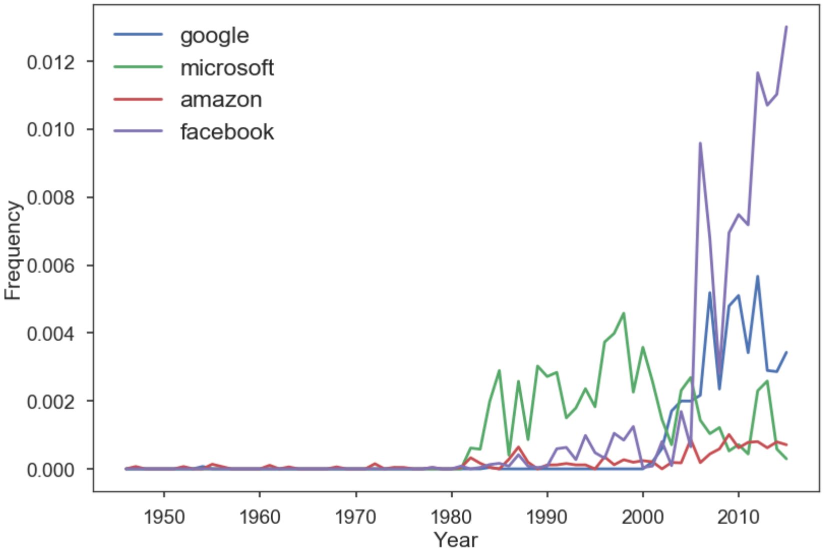


Figure 13: Relative frequency of various technology companies.

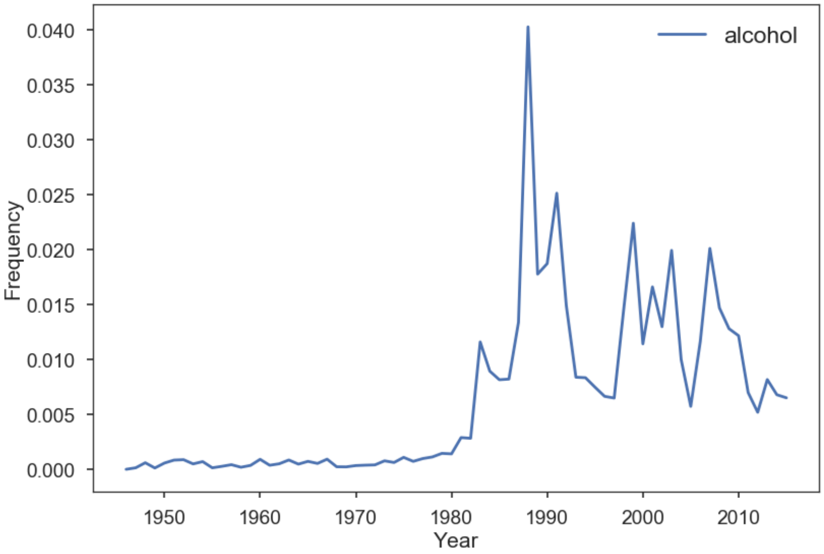


Figure 14: Relative frequency of the word 'alcohol'.

occurrences of the word 'girls'. We can perform additional analysis to determine whether there is indeed a positive correlation between the two words. To definitely prove or disprove any links, we would need to look through occurrences in the newspapers to see if the words are related in any way to each other.

Comparing the number of occurrences of the word 'bonfire' in Figure 17, we can see there

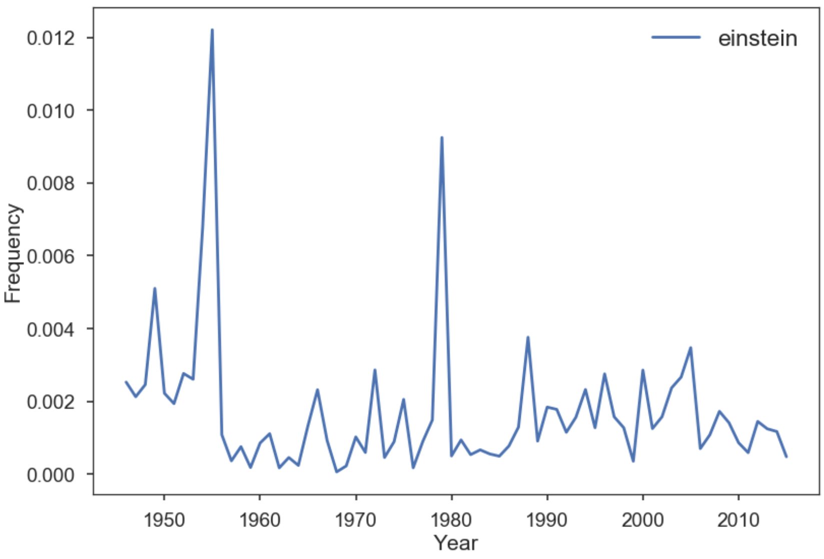


Figure 15: Relative frequency of the word ‘Einstein’.

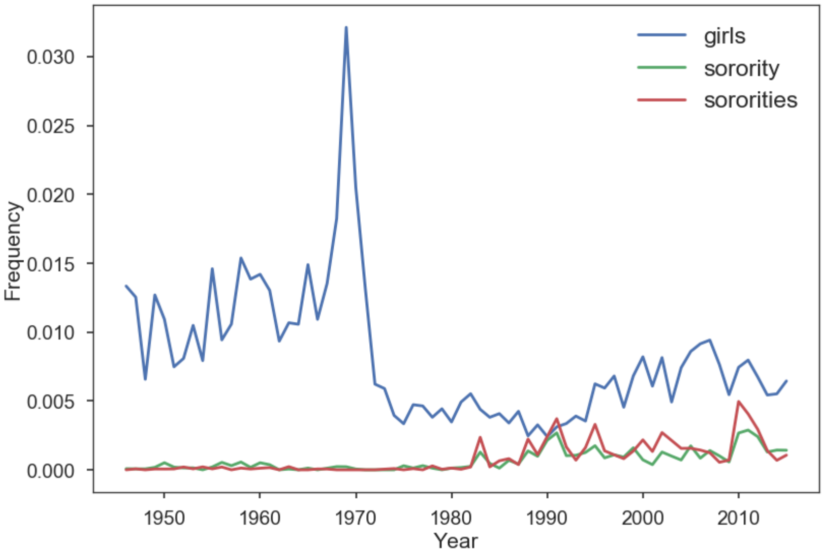


Figure 16: Relative frequency of the word ‘girls’ along with sororities.

is a large spike in the number of occurrences of the word in the years when Princeton holds a bonfire, that is beating both Harvard and Yale in the same year in football. Researchers can look for significant events which have happened over the years by looking for specific terms like these.

Comparing the number of occurrences of the word ‘deflation’ in Figure 18, we can see there was a large spike during the year of 2004 when the policy of Grade Deflation was announced and

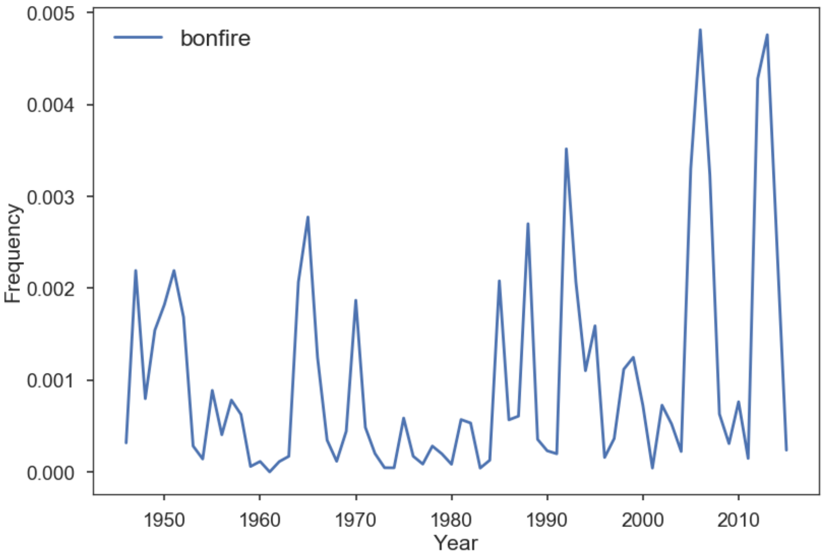


Figure 17: Relative frequency of the word 'bonfire'.

similarly in 2014 when it was officially removed. These tools allow researchers to visualize trends, especially historical events, for example, to track the history of Honor Code reform.

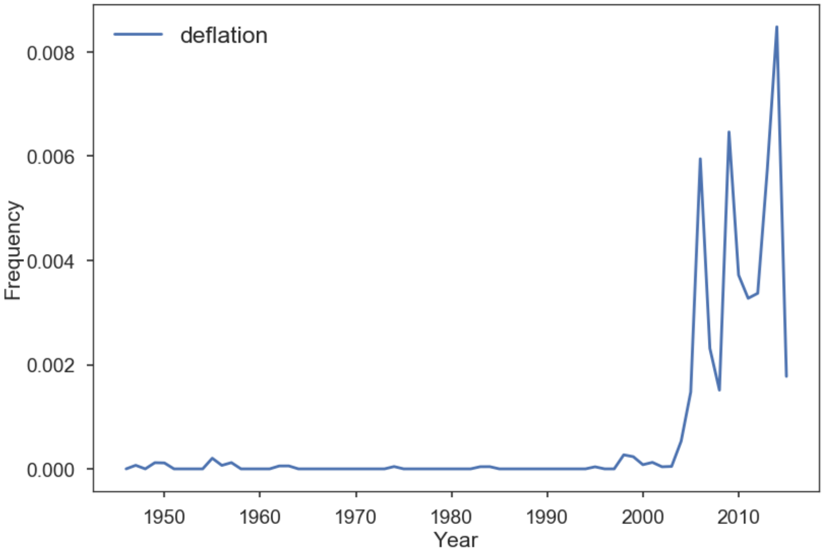


Figure 18: Relative frequency of the word 'deflation'.

Noteworthy mentions

For a consistency check, I looked for the most similar words to the month April. Unsurprisingly, March and February came at the top of the list, with cosine similarities of 0.9164 and 0.8961 respectively. The rest of the list was followed by other months, but I saw the strange word ‘jhday’ with a similarity score of 0.5725. I had never seen this word before, so I searched through the Daily Princetonian archives. Surely enough, it turned out that this was actually the word Friday, which had been written in fancy curly font which the OCR software consistently labeled ‘JHday’. (The curved top on the capital ‘F’ along with the ‘ri’ made it look similar to the capital letters ‘JH’).

I tried looking at the rise and fall of various departments at Princeton. However, when I searched for COS, I was surprised to find many occurrences of the three-letter word before computers were widespread. It turned out to be the first syllable of words such as cosmopolitan or costume. The words Amazon and Java also came into usage before the company and programming language came into existence, due to the Amazon river/rainforest and the island Java.

A similarity search for ‘math’ showed up ‘majors’ and ‘majoring’, which were both expected, but also several unexpected results including ‘ematics’, ‘profes’, ‘nomics’ and ‘chology’. This is probably due to the model replacing non-alphanumeric characters with whitespace, so the syllable-separated words of ‘math-ematics’, ‘profes-sor’, ‘eco-nomics’ and ‘psy-chology’ would be separated into two distinct words.