Exploring the Distribution of Word Frequencies Based on Their Semantic Properties

Eden Schaffer-Neitz
University at Buffalo
edenscha@buffalo.edu

Abstract

It's common knowledge that word frequencies in documents follow near-Zipfian (Zipf, 1935) distributions, that is, the frequency of a word is inversely correlated to its rank. What is lesser understood is whether or not the semantic qualities of words in a document follow this power law distribution. This project aims to visualize the distribution of semantic relationships between words in a document.

1 Introduction

Since Zipf (1935), we've been aware that word frequencies in a document, and in general, follow a Zipfian, or near-Zipfian, distribution. More specifically, the frequency of a word is inversely correlated to it's rank in the word frequency list, following the equation in Equation 1[1].

$$f(r) \propto \frac{1}{r^{\alpha}} \tag{1}$$

where the *r*th most frequent word scales according to 1.

What is lesser known, is how words cluster semantically. That is, if one was to take the embedding representations of words in a document and compare their pairwise similarities via some distance metric, how would these similarity scores cluster? At a higher level, do we tend to use words that are semantically similar to each other, or dissimilar, and if so, how similar?

Additionally, is this distribution the same for all parts of speech? Do adjectives cluster differently from nouns and verbs, or the same? How about across languages, do the distributions look the same across languages?

I plan to test the above questions in this project. Using Python and R, I'll calculate the pairwise cosine similarities of words in two documents and plot them according to their frequencies.

2 Phenomenon

The goal of this project is to answer the following questions:

- *How do words cluster when plotting the frequencies of cosine similarities between words in a document?*
- *Do the distributions depend on the part of speech of the words?*
- *Do these distributions change based on language?*

I believe the answers to these questions can be found experimentally. For each of these questions I have formulated my hypotheses.

Firstly, I believe the distribution will obviously be non-linear, particularly they will either follow a normal or inverse exponential distribution. I base this hypothesis on the assumption that we likely do not use words which are similar to each other more frequently than we use words dissimilar to one another.

My hypotheses to the second and third questions are that words will cluster the same regardless of part of speech and language. This is based on the same assumption as above, and these two factors do not affect this assumption. The final two questions are in place to verify the integrity of the findings in the first question. We do know that parts of speech and language affect word frequencies (Piantadosi, 2014), however it seems unlikely to me that these variables will affect the distributions at the semantic level.

Hopefully by the end of these experiments, we can conclude how the semantic qualities of words in a document are distributed, and more fully understand frequency distributions in natural language. I believe this work supplements that provided by Zipf (1936; 1949), and Piantadosi (2014).

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4176592/

3 Experiment

In order to carry out the experiments required to answer these questions I'll be utilizing Python to extract and process the relevant data from two books, *Frankenstein; Or, The Modern Prometheus*[2] by Mary W. Shelley in English, and *Les Trois Mousquetaires*[3] by Alexandre Dumas in French. I've chosen these two books for their popularity and languages, as well as their free-use license from Project Gutenberg [4].

After extracting and processing the data, I'll use R to plot the distributions and confirm or deny my hypotheses.

3.1 Methodology

At a high level, what I'll be doing is tokenizing and tagging text from books in two different languages. I'll tag these texts with their parts of speech. After this I'll then compute the similarity score between pre-trained multilingual GloVe (Pennington et al., 2014) vectors from Ferreira et al., 2016 using Cosine Distance, seen in Equation 2.

$$Cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| * ||\mathbf{y}||} \tag{2}$$

this equation computes the distance in N dimensional space where **x, y** are two vectors of length N. With these distances computed, I'll plot all points in R.

3.2 Procedure

In this section I'll detail every step taken in my experiment. The reader can follow along using the code found in Appendix A.

3.2.1 Python

In Python, I first import the necessary libraries and declare the file paths for the books I'll be using as well as the file paths for the pre-trained GloVe embeddings. I read these GloVe vectors into memory and initialize a dictionary for both English and French containing all of the K,V pairs, where K is A word in the vocab, and V is the embedding corresponding to word K.

After preparing the embedding dictionaries, I then read each book, creating a string containing the entire contents of the book in one. I tokenize this string into each individual word, and then tag each word with its corresponding part of speech. Tagging and tokenization is performed using the NLTK (Loper and Bird, 2002) libraries `word_tokenize`, and `averaged_perceptron_tagger`. I then remove the stopwords using NLTK's `stopwords` module from the tokenized and tagged words and return a list containing all of the processed tokens.

With this list of processed tokens, I continue by sorting each token by its part of speech, and return a list for each part of speech (Noun, adjective, and verb) containing the included tokens. Using these sorted lists, I iterate over each token in the list and search the relevant embedding dictionary for the embedding associated with each token. These embeddings are added to new lists (still sorted by part of speech and language) which are passed to the final step of processing the data, computing the pairwise cosine similarity scores.

For each embedding in each part of speech list I compute the cosine similarity score using Equation 2 between the current embedding and other embeddings in the list, the cosine similarity is not calculated between an embedding and itself. In the end, for each language (2) I return three lists, one for each part of speech I'm concerned with, for a total of six lists of cosine similarity scores.

Finally, I create a dictionary with six keys, one for each list *en_nouns, en_verbs, en_adjs, fr_nouns, fr_verbs, fr_adjs* and format this list into JSON (Pezoa et al., 2016) format for use in R.

3.2.2 R

Moving into R, I again first activate the required libraries, `rjson`[5] for using JSON in R, and `ggplot2` (Wickham, 2016) for plotting. I read the contents of the JSON file created in Python and assign it a variable.

Using this data I subset each list into a separate R vector. For each of these six vectors I randomly sample 10,000 cosine similarity scores and sort them in ascending order. I round each score to two decimal places, as a way to cluster scores which are different by less than two decimal places. Lastly I cast each of these vectors into a dataframe with two columns, the score value and the frequency of the score, and sort them by most frequent to least frequent.

Next, I plot each of these dataframes using `ggplot2` with Frequency on the Y-axis and the

[2] Sourced from: https://www.gutenberg.org/ebooks/84

[3] Sourced from: https://www.gutenberg.org/ebooks/13951

[4] https://www.gutenberg.org/policy/permission.html

[5] https://cran.r-project.org/web/packages/rjson/rjson.pdf

Language	Nouns	Verbs	Adjectives
English	0.26	0.29	0.28
French	0.12	0.12	0.10

Table 1: Average Cosine Score Values

corresponding cosine similiarty score on the X-axis. For example, one point could be *(0.23, 20)*. The X-axis is limited naturally to points within [-1, 1], as these are the bounds of the cosine similarity metric.

Finally, I compute the mean similarity score of each category in each language, for use in comparing the differences between the plots quantitatively.

4 Results and Discussion

As can be seen in Figures 1-6, the results of the experiment show that the distributions of each category are Gaussian or near-Gaussian. The position of the mean value on the X-axis varies slightly, and as seen in Table 1, the mean values of the English scores are much higher than the mean values of the French scores. The values between the parts of speech however, seem to remain relatively consistent throughout each language.

From these tables and figures, we can safely affirm my main hypotheses, that semantic representations between words will cluster normally. Likewise we can affirm the hypothesis that the lexical category of words makes little to no difference in the distribution of pairwise cosine similarity scores. However, it's clear by the data in Table 1 that language does affect the distribution, at least in this case, and my third hypotheses that language would not make a difference has been denied.

The normal distribution in these plots indicates that humans use words that are more semantically dissimilar than they do words that are similar. The averages in Table 1 show that the mean score is between 0.1-0.3, scores which are dissimilar on the scale of -1 to 1. Moreover, I believe that the variance between languages is interesting, and cannot at this time posit a hypothesis as to why there would be such a large difference between the average similarity scores. More information is likely required, such as more data, and data from different sources.

5 Conclusion

In conclusion, out of the three hypotheses,

- *How do words cluster when plotting the frequencies of cosine similarities between words in a document?*
- *Do the distributions depend on the part of speech of the words?*
- *Do these distributions change based on language?*

two were confirmed, while the third hypothesis relating to the language of the distribution was denied.

The confirmation of the first two hypotheses remains slightly uncertain without more data on a larger scale to back up the statistical significance of it. Adding more lexical categories and data sources to find a sufficiently large amount of data points could lead to a distribution with variance high enough to consider them independent of one another.

In order to more fully understand the cause of the differences in languages, I believe more work is required. The data could be an issue, seeing as how they came from books in different time periods, with different authors, about different topics, would a larger and more diverse dataset provide the same results? Likewise, would adding more languages to the comparison show a similar pattern, with significant differences between their average similarity scores, or is the French and English case an outlier? Ideally I'd like to perform an ANOVA test on these new findings to determine whether there is a statistically significant contrast between the distributions across languages.

My hope is that this project enlightens people on some of the statistical properties of distributed semantics, and encourages future work to uncover more.

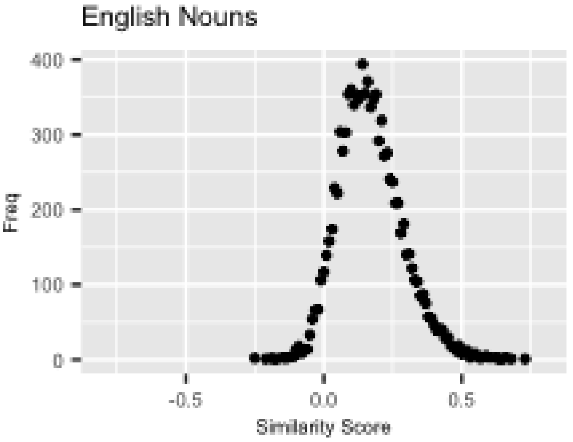

Figure 1: English Nouns

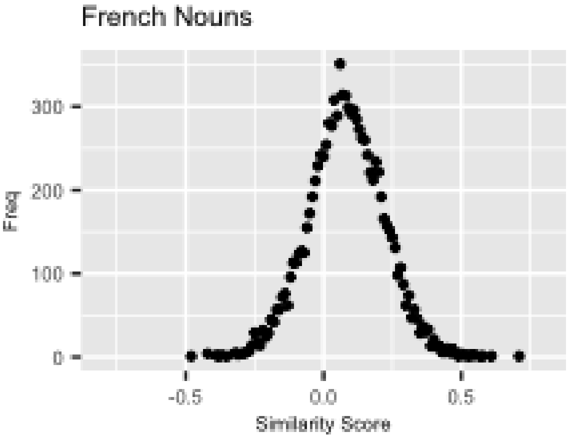

Figure 4: French Nouns

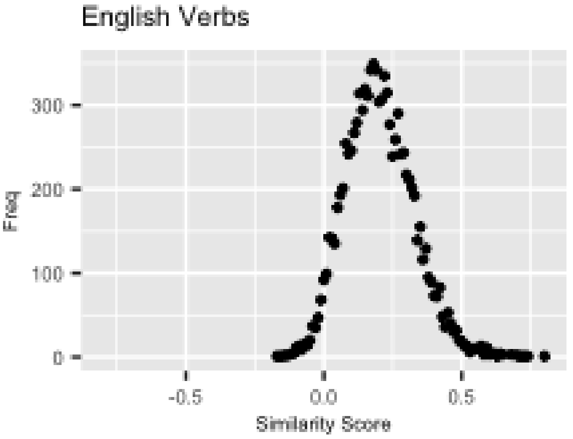

Figure 2: English Verbs

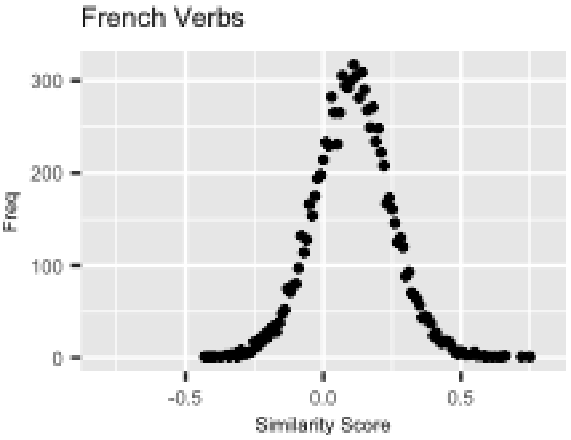

Figure 5: French Verbs

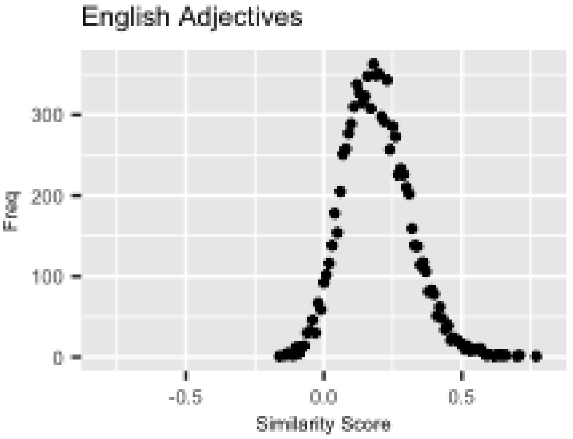

Figure 3: English Adjectives

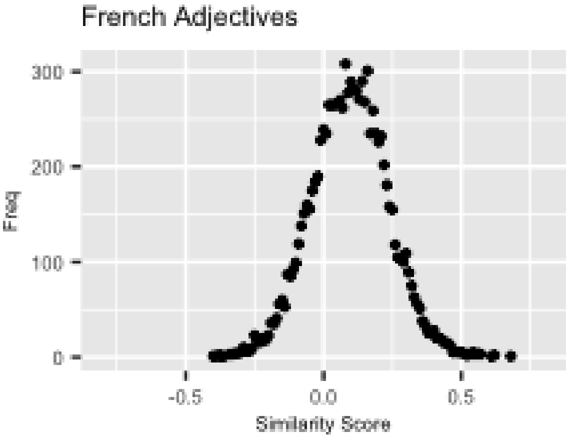

Figure 6: French Adjectives

References

Daniel C. Ferreira, André F. Martins, and Mariana S. Almeida. 2016. Jointly learning to embed and predict with multiple languages. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Felipe Pezoa, Juan L Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. 2016. Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, pages 263–273. International World Wide Web Conferences Steering Committee.

Steven T. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin amp; Review*, 21(5):1112–1130.

Hadley Wickham. 2016. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

George Zipf. 1935. *The Psychobiology of Language: An Introduction to Dynamic Philology*. M.I.T. Press, Cambridge, Mass.

George Kingsley Zipf. 1936. *The psycho-biology of language: An introduction to dynamic philology*. Routledge.

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.

A Appendix

All code can be found on GitHub at: https://github.com/eschaffn/514-Final-Project.