A TITLE ABOUT WORD EMBEDDINGS

Mike Burnham
Department of Political Science
Penn State University
mlb6496@psu.edu

December 18, 2019

ABSTRACT

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, acvenenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus conguequam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nullafacilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittisdui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna.Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabiturconsectetuer

1 Introduction

This paper examines the use of word embeddings for both the qualitative and quantitative research of ideologies. Word embeddings are language models primarily used to boost the performance of machine learning algorithms on natural language processing (NLP) tasks. They have been wildly successful in this arena because of their ability to capture the many dimensions of semantics in language. For this same reason word embeddings can be used not only as a data source fed to predictive algorithms, but as a standalone model from which social scientists can conduct descriptive and inferential research.

More specifically, I have three primary objectives: First, I justify this research agenda by briefly reviewing current methods used to leverage text as data, and then examine what exactly word embeddings are and how they alleviate some shortcomings of current techniques. Second, I show that word embeddings can be leveraged to accomplish several of the same tasks dictionary and word count methods are currently used for namely qualitative or descriptive analysis and scaling or positional estimates of ideological stances. I argue that word embeddings accomplish this in a more robust and theoretically sound manner. Finally, I discuss a research agenda moving forward to explore the most rigorous and practical implementation of word embeddings in social science research.

2 Literature Review

In this section I first briefly review the literature on what techniques political scientists currently use to leverage text as data. I then give a brief overview of what word embeddings are and the theoretical validation for their use. Finally, I examine some of the recent developments in leveraging word embeddings that I hope to build on with this research.

2.1 Classification and scaling

Social scientists have largely relied on what is known as a bag-of-words method for NLP tasks. Bag of words methods analyze text by treating documents as unorganized collections of words, counting the instances of words within a text, and using those word counts to infer meaning or make predictions [Monroe and Schrodt, 2008]. Counted words may be part of a pre-compiled list of significant terms such as the Dictionary of Affect in Language [Whissell, 2009] or Linguistic Inquiry and Word Count [Pennebaker et al., 2001] which are used to count words associated with sentiment or certain emotions, or they may follow some unsupervised algorithm that assigns weights to all words and analyzes text best and the words and their associated weights that appear in a text. Wordfish and Wordscores are two commonly used pieces of software that rely on a bag-of-words approach to classify documents [Proksch and Slapin, 2008, Laver et al., 2003].

The primary shortcoming of this method is that it ignores context and assumes each word is an independent observation [Bruinsma and Gemenis, 2017, Monroe and Schrodt, 2008]. The phrase not bad, for example, could mean anything from mild criticism to strong praise depending on the context in which it is used. Yet, a bag of words approach only sees the words not and bad which, if doing sentiment analysis, will always count as two negative words within the document. This is somewhat alleviated by whats known as n-grams, or a window of n words that treats all words within that window as a single observation. However, n-grams still ignore larger sentence structure, multiply the number of unique features in your data set by n, and maintains the naive assumption that there is no relationship between features.

These shortcomings mean that bag-of-words approaches tend to be noisy. Wordscores, for example, has a tendency to overweight high-frequency words and does not account well for sampling variation [Monroe et al., 2008, Lowe, 2008]. Wordfish, on the other hand, suffers from interpretability challenges. Because the algorithm is unsupervised and places documents on a single dimension that is defined by the algorithm rather than the researcher, it is not necessarily the case that the dimension is related to the topic of interest. Wordfish may capture sentiment rather than party affiliation, for example. Because both methods depend on word frequencies, both techniques struggle with unique or low frequency words even though low frequency words may be particularly informative to a human reader. In modern applications, bag-of-words approaches are used because they are computationally cheap. They are rarely, if ever, the most robust approach to an NLP task.

2.2 Word embeddings and the distributional hypothesis

Popularized in 2013 by Googles Word2vec algorithm, word embeddings greatly alleviate many shortcomings of the bag-of-words approach [Mikolov et al., 2013]. Speaking broadly, word embeddings are language models in which each word is represented by a vector of numbers. The most basic algorithm works by observing a corpus of documents and noting which words appear together. With enough observations, a predictive model is able to determine which word is most likely to appear based on its surrounding words, or vice versa. The series of numbers that is fed to this model to predict a word is known as a word vector [Goldberg and Levy, 2014]. Each word in the data set has a word vector representation, and the collection of these word vectors is the word embedding.

Word embeddings have proven particularly adept at modeling language and form the backbone of modern NLP. Word vectors are able to capture sentiment, part of speech, or even abstract concepts associated with a word. Thus, word vectors do not just represent a word, but the conceptual and semantic ideas a word represents. A famous example is that the word vector for queen is roughly equal to the vector for king, minus the vector for man, plus the vector for woman. That is:

$$Queen = KingMan + Woman \tag{1}$$

While this example serves little practical purpose, it does demonstrate surprising capacity of embeddings. To explain further, word embeddings are able to recognize that the concept of king is defined by a political position and the male gender. If I take the vector of numbers that represents king, subtract the vector for man, add the vector for woman, the resulting vector is defined by political position and the female gender. As expected, the nearest neighbor to my new vector is the vector that represents queen. Numeric representations of semantic concepts allow us to plot them in high dimensional space, calculate the distance or similarity between concepts, and even preform mathematical operations with them.

2.3 Use of word embeddings to measure semantic differences

Within industry and academia, word embeddings are primarily used to boost the performance of machine learning algorithms on natural language processing tasks. This trend has largely carried over in to political science. Rudkowsky et al., for example, used word embeddings to boost the performance of an algorithm that labels the sentiment of Austrian parliamentary speeches [Rudkowsky et al., 2018]. Likewise, Iyyer et al. use word embeddings and recursive neural networks to assign binary ideological labels to political text [Iyyer et al., 2014]. Using embeddings to boost the performance of algorithms is certainly an important and worthy area of research. It is particularly useful in political science when categorical labels such as party affiliation or sentiment need to be estimated for a data set.

However, word embeddings are rich in semantic information and are worth examination as a language model themselves rather than being relegated solely to feeding algorithms. The foundation of word embeddings is known as the distributional hypothesis. The distributional hypothesis states that words appearing in similar contexts have similar meanings [Goldberg and Levy, 2014, Mikolov et al., 2013]. Intuitively, this makes sense synonyms such as table and desk are likely to appear next to similar words. Therefore, their numeric vectors will be similar and if we were to calculate the distance between these vectors, table would be closer to desk than queen.

For our purposes, the strength of word embeddings is directly tied to the distributional hypothesis. If the hypothesis holds, the converse is also true. Namely, that dissimilar words are found in dissimilar contexts. By extension, if two groups use the same word in different contexts, the two groups are assigning different meanings to the word. These differences in meanings can be used to represent ideological differences. For example, a word embedding training on conservative texts may find that the word vector for abortion is mathematically closer words like murder and baby than it is to words like woman and choice while the converse is true for an embedding trained on liberal texts. These mathematical differences between word vectors represent ideological fault lines between groups.

In recent years, two groups of researchers made significant strides with using word embeddings for measuring semantic differences. Hamilton et al. leveraged word embeddings to examine semantic changes across the dimension of time [Hamilton et al., 2016]. Their technique uses text associated with specific time periods to create embeddings across time. Because different word embeddings occupy different vector spaces however, it is impossible to calculate distance between words directly. To get around this, the authors utilize orthogonal procrustes analysis which, roughly speaking, finds a matrix that most closely aligns the two embeddings, allowing them to compare across vector space. In doing so, they successfully demonstrated the evolution of a words semantic meaning across time.

Azarbonyad et al. advanced this technique by replacing the dimension of time with viewpoints. To test the method, they used text from the UK parliament to examine words to examine words where they expected ideological convergence and divergence [Azarbonyad et al., 2017]. In both instances they found that word vectors diverged where ideological differences were expected and converged on words where no differences were expected.

3 Research Goals and Design

My first goal is to reproduce results similar to the qualitative ones of Azarbonyad et al. that extracted descriptive words for ideological spaces. However, I attempt this in a single vector space rather than across vector spaces. This offers some potential advantages in both ease of implementation and interpretability as it foregoes the procrustes transformation. Second, I test whether word embeddings can be used for ideological position estimates, similar to what is currently done with bag-of-words techniques such as wordfish. More specifically, I evaluate the use of cosine similarity and relative cosine similarity as reliable measurements of distance between ideological positions. This leads me to my three hypotheses, one qualitative and two quantitative:

Hypothesis 1: The words in closest proximity to ideologically significant words will highlight points of contention or agreement between groups and can be used for descriptive analysis.

Hypothesis 2: If text is divided into two groups and separate word vectors for the same word are calculated between the two groups, that word vector will have a lower cosine similarity with its counterpart if the text is labeled by ideological group rather than labeled randomly.

Hypothesis 3: If separate word vectors for the same word are calculated for ideologically distinct groups, word vectors that represent points of disagreement will have a lower cosine similarity than those that represent points of agreement.

In the following sections, I first discuss my research design and data used to test these hypotheses in more detail.

3.1 Data

For this research, I use twitter data from incumbent Republican and Democratic congress members between November 6, 2019 and December 17, 2019. Tweets were collected daily via twitters resting API to compile a comprehensive data set of all tweets in this time period. This data is suited to this research for a number of reasons:

1. There are unambiguous ideological boundaries or teams. Because this method does not specify an ideological scale a priori, it is fundamentally referential and a test set with clearly demarcated borders is ideal.
2. Data is generated within a single semantic environment where all users follow the same rules and restrictions.
3. It provides a highly reproducible test case that is publicly available and can potentially be updated in real time to see if assumptions hold across time and changes in the news cycle.

All tweets used will be originally generated from the specified user (i.e. no retweets, quotes, etc.) in order to avoid redundancy in the data. In total I collected roughly 28,000 tweets. The distribution by party is shown in figure 1.

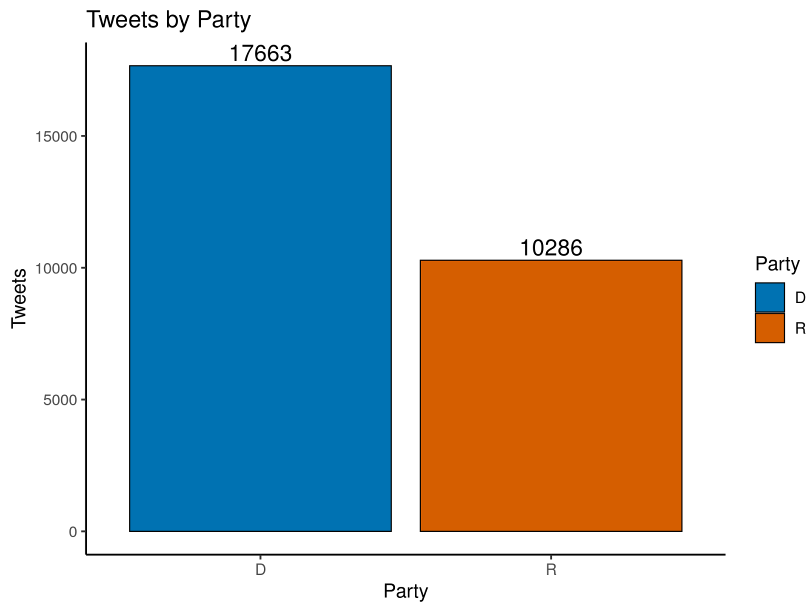

Figure 1: Distribution of tweets

In conjunction with the text from congress members, I compiled a list of ideologically significant words that represent points of agreement and disagreements between the parties, as well as a list of ideologically neutral words. To generate this list of words, I first created a list of 100 politically relevant words. I then narrowed this list down to words that have unambiguous meaning, represent clear ideological divides or agreement, and are used more than ten times by each party. For example, impeach is a good word for my purposes because of its clear meaning, high usage, and both parties have clear, unified positions. In addition to the ideologically significant words, I compiled a list of ideologically neutral words that can be found in any context. A word count for each list can be found in table X and a list of these words if found in the appendix.

Category	*n*
Disagree	29
Agree	10
Base	47

Table 1: Key words by category

3.2 Pre-processing and cleaning

To clean the text I removed websites, emoji, punctuation, and digits via regular expressions. I used spaCys language model to tokenize, lemmatize, and remove stop words from the text [Honnibal and Johnson, 2015]. The stop words removed consist of spaCys default list with minor alterations to accommodate for specifics in the data set. For example, the word make was removed from the list because of its significance to the phrase make America great again. A list of stop words is located in the appendix.

Next, I isolate and tag a single word from the aforementioned list of ideologically significant words. This process involves several steps, the first of which is combining synonyms into a single term so that a single word vector can be calculated for a concept. A potential criticism of this process is that by replacing synonyms of words I am compromising some of the semantic meaning of these words. I offer two defenses: First, in all cases the replacement of synonyms into a single term is minor and unambiguous. Examples include replacing President Trumps twitter handle @realdonaldtrump with simply trump or replacing the term leftist with liberal. Second, differences in terminology primarily reside on ideological fault lines. Because different word vectors are calculated for each ideological group the semantic differences represented by these slight changes in terminology will still largely be represented in the single word vectors and their associated cosine similarity scores. A dictionary of words and their associated synonyms if found in the appendix.

The next step isolates each instance of the key word by inserting spaces between these instances. For example, the string "#supporttrump" would be altered to "support trump" after the initial cleaning and key word isolation. This allows the word embedding model to treat support and "trump" as separate tokens.

Once key words are isolated, final step is to tag each word with either a political group or randomly assigned label depending on which test I am conduction. If the string "#supporttrump" was tweeted by a republican, it becomes "support trump_r." After this process is complete I can calculate a word vector for both trump_r and trump_d which represent the Republican and Democratic use of the word trump respectively.

I use Word2vec with a skip-gram architecture as implemented in the python Gensim library to construct embeddings [Řehůřek and Sojka, 2010]. Embeddings are trained with word vectors of 100 dimensions and a window size of 10. A complete list of hyper-parameters used to train the model is found in the appendix.

4 Results

4.1 Hypothesis 1: The words in closest proximity to ideologically significant words will highlight points of contention or agreement between groups and can be used for descriptive analysis.

To test my first hypothesis I examine a list of the ten most similar words as measured by cosine similarity for each word, and a similar list for that words counterpart from the other party. I expect that words unique to each list represent ideological divides between the parties, and common words represent points of agreement. As cosine similarity between a word and its counterpart increases, I expect that overlap between the two lists will increase.

From the disagree set of words I used the word impeach because of its relevancy to the news in the time period I collected the data. Because word vectors within the embedding contain 100 dimensions it is impossible to plot them precisely. To compensate, I used principal component analysis (PCA) to reduce the number of dimensions. PCA preforms a linear transformation on the word vectors that combines variables into n principal components that explain the maximum amount of variance. Because each component is an amalgamation of the other dimensions within the word vector, components are not readily interpretable. Rather, this technique is solely used to enable a rough visualization of the datas separation. Figure X below shows the scatter plot for the word impeach and the ten most similar words by party after applying PCA to reduce the dimensionality down to three. With three dimensional PCA I am able to explain roughly 75% of the variance between word vectors, meaning that while the plot is not an exact visualization of the distance between words, it is a decent approximation.

The plot for impeach shows two distinct clouds for the parties with little overlap. As shown in table X, the parties have no common terms among the ten terms most similar to their version of impeach. More significantly for this hypothesis, the most similar terms clearly reflect party positions. Democrat words feature terms such as abuseofpower and exposethetruth while Republican words include partisan, charade, and circus.

I repeated this process for the word robocall due to bipartisan legislation to crack down on robocalls during the period of data collection. Figure X and table X show the opposite of impeach with significant overlap between the two versions of robocall. Most telling is that one of the closest words to the Democrats version of robocall is the republicans version, suggesting the words are synonymous to some degree. Finally, the neutral word look exhibits what appears to be largely noise. The two words are largely synonymous, but the nearest neighbors to each words do not appear to carry significant ideological meaning.

4.2 Hypothesis 2: If text is divided into two groups and separate word vectors for the same word are calculated between the two groups, that word vector will have a lower cosine similarity with its counterpart if the text is labeled by ideological group rather than labeled randomly.

The second hypothesis is designed to test if word embeddings capture semantic differences between groups and if cosine similarity can be used to quantify those differences. Generally speaking, my expectation is that all words will have a lower cosine similarity with its counterpart when labeled by party rather than randomly, albeit I do expect the difference between party and randomly labeled data to be greatest between words that represent disagreement and smallest between words that represent agreement. I expect near universal difference because even if a word represents a point of agreement between the parties, I still expect slight variation in the partys talking points. There may be exceptions to this, for example if the two parties were to coordinate a public relations campaign together.

To test this hypothesis, I conducted a permutation test on individual words. I first label each text according to its associated party, and then calculate the observed cosine similarity between a single word and its counterpart. I then randomly permute the text labels, recalculate cosine similarity, and repeat this process 1,000 times. From here I can determine probability that my observed cosine similarity is drawn from my data set of cosine similarities calculated with randomized labels. A low p-value indicates that text labels do matter, and that the word embedding is capturing semantic differences between the groups. Because this test takes about 14 hours of computational time to complete, I repeated it for only one word from each of my three categories. For the agree and disagree categories I chose usmca and trump respectively due to their relevance to the news cycle during the period I collected data. A word from the base word list was chosen at random. In this case I used place.

Figures X show a histogram of the results

4.3 Hypothesis 3: If separate word vectors are calculated for ideologically distinct groups, the word vectors that represent points of disagreement will have a lower cosine similarity than those that represent points of agreement.

My final hypothesis seeks to test the generalizability of the results from the second hypothesis. To do so, I calculated the cosine similarity for each word and its ideological counterpart in my list. I then calculate the mean cosine similarity for each of the three word groups. Namely, words that represent disagreement, agreement, and non-ideological words. I use a difference of means test to evaluate differences between the groups. I expect that words in the disagree category will have the lowest cosine similarity, the agree will have the highest, and non-ideological will be in the middle. Because the number of words is relatively small, I use a permutation test and bootstrap confidence interval to supplement my results.

There are two primary factors driving these inconclusive results. First are potential shortcomings in my list of terms. For example, I selected the word police under the expectation that it would capture different attitudes on police brutality and the Black Lives Matter movement. Generally speaking however, elected officials seemed reluctant to criticize police on social media. Ongoing police brutality surrounding the Hong Kong protests has been denounced by both parties on social media. Police therefore did not capture the intended ideological dimensions. Similarly, the word healthcare which was included in the disagree list was probably more noise than signal. While there is certainly disagreement between the parties on healthcare, clear party platforms have not emerged and there is significant disagreement within the parties as well. Thus, I would probably need to draw different ideological boundaries to measure this particular concept.

The second shortcoming is simply that cosine similarity is a bit of a blunt instrument. In their research on synonym extraction, Leeuwenberg et al. showed that cosine similarity alone is a bad indicator to determine if two words are synonymous. Instead, they propose a technique they call relative cosine similarity which examines the similarity between two words relative to other words in the corpus. As a rule of thumb, if relative cosine similarity is greater than 0.10, the two words are more similar than an arbitrary word pair. Accordingly, I repeat the above analysis using relative cosine similarity and present the results in the table X.

Relative cosine similarity reinforces the above findings and appears to give more robust results than simple cosine similarity. The strongest indicator of this is the significant jump in similarity among the base words. While some deviation is to be expected because Democrats and Republicans do not engage in the same topics of discussion with the same frequency, I expect words that can be found in any context and generally carry no ideological significance to be relatively synonymous.

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

> Hasselmo, et al. (1995) investigated…

`https://www.ctan.org/pkg/booktabs`

4.4 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas

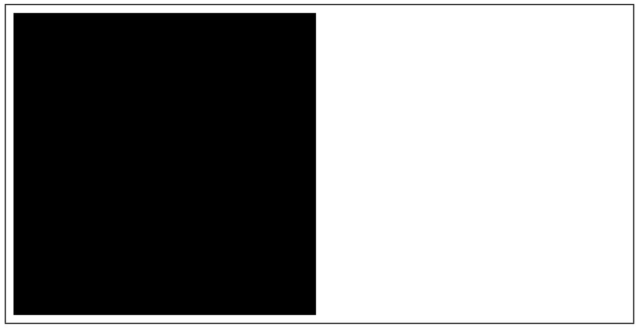

Figure 2: Sample figure caption.

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Table 2: Sample table title

imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

See Figure 2. Here is how you add footnotes. [1] Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

4.5 Tables

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

See awesome Table 2.

4.6 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

References

[Azarbonyad et al., 2017] Azarbonyad, H., Dehghani, M., Beelen, K., Arkut, A., Marx, M., and Kamps, J. (2017). Words are malleable: Computing semantic shifts in political and media discourse. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1509–1518. ACM.

[Bruinsma and Gemenis, 2017] Bruinsma, B. and Gemenis, K. (2017). Validating wordscores. *arXiv preprint arXiv:1707.04737*.

[Goldberg and Levy, 2014] Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

[1] Sample of the first footnote.

[Hamilton et al., 2016] Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*.

[Honnibal and Johnson, 2015] Honnibal, M. and Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

[Iyyer et al., 2014] Iyyer, M., Enns, P., Boyd-Graber, J., and Resnik, P. (2014). Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122.

[Laver et al., 2003] Laver, M., Benoit, K., and Garry, J. (2003). Extracting policy positions from political texts using words as data. *American political science review*, 97(2):311–331.

[Lowe, 2008] Lowe, W. (2008). Understanding wordscores. *Political Analysis*, 16(4):356–371.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Monroe et al., 2008] Monroe, B. L., Colaresi, M. P., and Quinn, K. M. (2008). Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

[Monroe and Schrodt, 2008] Monroe, B. L. and Schrodt, P. A. (2008). Introduction to the special issue: The statistical analysis of political text. *Political Analysis*, 16(4):351–355.

[Pennebaker et al., 2001] Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

[Proksch and Slapin, 2008] Proksch, S.-O. and Slapin, J. B. (2008). Wordfish: Scaling software for estimating political positions from texts. *Version*, 1:323–344.

[Řehůřek and Sojka, 2010] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

[Rudkowsky et al., 2018] Rudkowsky, E., Haselmayer, M., Wastian, M., Jenny, M., Emrich, Š., and Sedlmair, M. (2018). More than bags of words: Sentiment analysis with word embeddings. *Communication Methods and Measures*, 12(2-3):140–157.

[Whissell, 2009] Whissell, C. (2009). Using the revised dictionary of affect in language to quantify the emotional undertones of samples of natural language. *Psychological reports*, 105(2):509–521.