A Timely Intervention: Tracking the Changing Meanings of Political Concepts with Word Vectors

Emma Rodman (erodman@uw.edu)
Department of Political Science, University of Washington

Forthcoming in *Political Analysis*. Please cite published version.

Abstract

Word vectorization is an emerging text-as-data method that shows great promise for automating the analysis of semantics – here, the cultural meanings of words – in large volumes of text. Yet successes with this method have largely been confined to massive corpora where the meanings of words are presumed to be fixed. In political science applications, however, many corpora are comparatively small and many interesting questions hinge on the recognition that meaning changes over time. Together, these two facts raise vexing methodological challenges. Can word vectors trace the changing cultural meanings of words in typical small corpora use cases? I test four time-sensitive implementations of word vectors (`word2vec`) against a gold standard developed from a modest dataset of 161 years of newspaper coverage. I find that one implementation method clearly outperforms the others in matching human assessments of how public dialogues around equality in America have changed over time. In addition, I suggest best practices for using `word2vec` to study small corpora for time series questions, including bootstrap resampling of documents and pre-training of vectors. I close by showing that `word2vec` allows granular analysis of the changing meaning of words, an advance over other common text-as-data methods for semantic research questions.

1 Introduction

Language changes over time, reflecting shifts in politics, society, and culture. The word "apple" used to refer just to a fruit but now also refers to a company; the word "gay" used to refer to a mood or personality type but now almost solely refers to a sexual orientation. Similarly, the meaning of words like "freedom" or "citizenship" have changed dramatically over time, reflecting the political and cultural shifts in a given society (Foner 1998; Goldman and Perry 2002).[1] The diachronic study of language – the study of changes in language over time – tracks these cultural

Author's note: Replication materials for this paper are available (Rodman 2019).

[1] Political scientists tend to be particularly interested in changes in the broad semantics of words which we might describe as essentially contested and continually evolving, like freedom, citizenship, equality, peace, or rights (Gallie 2013), or nouns like those denoting party or ideology (i.e. Republican, Democrat, liberal, or conservative).

shifts in a word's meaning, encapsulating not merely the word's dictionary definitions (which may not shift) but also its central referents, contexts of use, associated sentiment, and typical users, which together reflect political attitudes and culture (Hamilton, Leskovec and Jurafsky 2016*a*).

Studying changes in language over time, as a window into a specific political and cultural context, rests on the assumption that words can have many layers of meaning. Producing a thick understanding of this semantic architecture of words (de Bolla 2013) and languages has been a central goal of 20th century linguistics theory and the computational linguistics methods that have sought to implement those theories. While many computational methods have been produced to map and measure semantics (e.g. latent semantic analysis and semantic folding, among others), word vectorization methods (also called word embeddings) built on shallow neural networks have recently emerged as exciting new front runners in this effort. Recently developed algorithms like `word2vec` and others (Mikolov, Yih and Zweig 2013; Pennington, Socher and Manning 2014) have made this word vector modeling much more computationally accessible to practitioners, expanding efforts to apply word vectors to many text-based questions.

These word vectorization methods are unsupervised methods, in that they take no human inputs aside from the corpus of texts and a few model hyperparameter settings. Relying on the semantic information intrinsic to the collocation of words in the texts, these models produce low-dimensional vectors that represent each word. These word vectors have been shown to encode a tremendous wealth of semantic information. For instance, scholars have shown that we can use vector representations of words to accurately answer analogy questions like Man is to King as Woman is to _____ (Mikolov, Yih and Zweig 2013). By starting with the vector for the word "king", subtracting the vector for the word "man", and adding the vector for the word "woman," you end up positioned in the model space closest to the vector for the word "queen." The changing proximity of words to one another in these model spaces has also shown a remarkable capacity to capture semantic relationships and cultural-temporal shifts in the architecture of words (Mikolov, Sutskever, Chen, Corrado and Dean 2013; Hamilton, Leskovec and Jurafsky 2016*a*). Recent work, for instance, has used word vectors to track the changing cultural meanings and stereotypes associated with race, ethnicity, and gender (Garg et al. 2018).

In this paper, I utilize a substantive example from my own work as a scholar of American political thought to demonstrate both the utility and empirical validity of word vector models for questions of semantic change over time. I argue that understanding contemporary American political culture requires an understanding of how the architecture of the word "equality" has changed over time. Rather than projecting contemporary meanings of the word backward onto the past, this requires a genealogy of the semantics of American equality, excavating what equality meant to the people who used it in any given historical epoch. As part of that analysis, I applied

computational text-as-data methods to help me make sense of uses of equality in a sizeable corpus of historical American newspapers. Because the goal was to track the semantics of a single word, word vectors appeared promising as a method to study equality's changing historical meanings; indeed, as I show in section 7, there are questions of interest which would not be answerable with other common approaches to studying text as data.

However, nearly all work on word vectors – most of which has been done by computer scientists – trains the models that produce these vectors on extremely large corpora of texts, containing billions of words.[2] Much of this work also looks at only a single moment in time, attempting to replicate the broad semantic structure of a language. By contrast, the data set of texts used in this paper ($n = 3,105$) contains a total of only 206,190 words, and I model change across the period spanning 1855 to 2016 in a specific, relatively small subset of English language newspaper texts. In general, political scientists will be interested in comparing the semantics of smaller collections of texts (say, for instance, exploring the variation in press releases from different sets of elected officials or showing how the language in governance documents related to climate change changes over time).

Although computer scientists and computational linguists have developed several sophisticated time-sensitive implementations of word vectorization (Kim et al. 2014; Kulkarni et al. 2015; Hamilton, Leskovec and Jurafsky 2016*b*; Yao et al. 2018; Bamler and Mandt 2017), their efforts to date have done little to help guide practitioners, for several reasons. First, existing studies on how to use word vectors for diachronic analysis have been performed on extremely large corpora, often far exceeding typical use cases for political scientists.[3] While the literature offers some guidance on using word vectors with smaller corpora more generally (Antoniak and Mimno 2018), there has been no integration of these developments into diachronic questions, with the result that practitioners cannot be sure whether any of the existing diachronic methods will produce valid results when used on more modest corpora. Second, little explicit guidance is offered as to the appropriateness or details of a given time-sensitive implementation for given use cases. The challenges, hurdles, and occasionally even implementation details of diachronic analyses using word vectors have been left largely implicit in existing studies, leaving new users without the necessary introductory information to fully understand the method's structure, reasonability, or utility for a given task.[4]

[2]For instance, the landmark Mikolov, Yih and Zweig 2013 paper used a Google News corpus with six billion tokens (roughly, words) and a vocabulary size of one million words. Several diachronic analyses (Hamilton, Leskovec and Jurafsky 2016*b*; Kim et al. 2014) utilize the Google N-Gram data set, which is constructed from 6% of all books ever published and contains $8.5 * 10^{11}$ tokens. Hamilton et al.'s smallest corpus, a genre-balanced collection of literature, contains 410 million tokens.

[3]Although no hard guidelines exist about how much text is needed to produce quality embeddings, the general consensus is that more text is better. As I discuss throughout the paper, however, validating models and constructing them with small corpus best practices is a more useful way of thinking about the small corpus problem than arbitrary benchmarks for corpus size.

[4]Work on diachronic word vectors is fast moving and ongoing. The current frontier of work on time-sensitive word vector models is dynamic embedding models like those produced by Yao et al. 2018 and Bamler and Mandt 2017, as

After introducing first principles of word vectors, this study moves to fill those gaps by 1) articulating both the promise and the challenges of diachronic word vectorization models, 2) describing four possible diachronic implementations of the method for political scientists with modest programming skills in Python, and 3) offering an empirical assessment of the success of these time-sensitive formulations of word vector models on a relatively small corpus. Though larger corpora are generally preferred for word vector models, I suggest here that empirical validation of small corpus model results should substitute general pessimism about the utility of small corpus vector models. For my empirical assessment, I used human-coded documents and supervised topic modeling on my newspaper corpus to produce a gold standard – known semantic relationships between "equality" and other words like "race" and "gender" – and then tested four methods of time-sensitive word vectorization to see which method reproduced the gold standard semantic relationships most closely.[5] My results are encouraging, suggesting that with attention to new methods like bootstrap resampling of documents and pre-training of word vectors, even relatively small corpora can be successfully modeled using word vectorization. By empirically validating best practices for political science practitioners, I seek to encourage the use of word vectorization methods in political science, which – despite showing great promise as a method for questions of interest to political scientists – has lagged economics and other social sciences in uptake (Gurciullo and Mikhaylov 2017).[6]

I begin by discussing the utility of text-as-data methods – particularly word vectorization or word embeddings – for typical theoretical questions of interest to political scientists (§2). I then turn to the mechanics of word vectors, offering an introduction to distributional semantics, followed by word vectorization theory, methods, and implementations in common programming languages (§3). I then describe the challenges such methods present to analysts interested in studying semantic changes over time (§4). Using a dataset of newspaper articles which has been previously topic coded and modeled using supervised topic modeling (§5), I propose and empirically test four solutions to the challenges of temporal analysis using word vectors and I report the results. I find that there are certain best practices for using word vectorization methods to answer questions about changes in word meaning over time, including bootstrap resampling of documents, selective language stabilization, and chronological training (§6). Finally, I show the utility of this validated word vectorization model for political scientists by demonstrating its ability to answer questions about the changing meaning of words over time that other common text-as-data methods are

well as models that use convolutional neural networks (Kim 2014; Zhang and Wallace 2015). While showing much promise, they are not tested in this paper because their computational demands, the opacity of the details of their implementation, and their complexity make them currently inaccessible to typical political science users. This is likely to change quickly, however, and those interested in word vector models should keep their eye on developments in dynamic embeddings.

[5]Supervised topic modeling, as I describe at length in section 5, takes an input of human-coded documents (a training set) which the computer then uses to learn the topics and then apply those learned topics to the rest of the corpus of documents.

[6]Notable exceptions that draw on political examples or seek to contribute to ongoing substantive debates in the political science literature include Iyyer et al. 2014, Nay 2017, Gurciullo and Mikhaylov 2017, and Rudkowsky et al. 2018. To my knowledge, however, no studies have yet been published in political science journals.

unable to answer (§7).

2 Studying Words

Political scientists interested in theory have begun to recognize the utility of computational text-as-data methods, particularly for questions that involve large scale text analyses or comparisons across divergent cultural contexts. For instance, in their recent work topic modeling medieval political theory texts that offer advice to princes and sultans in the Islamic and Christian worlds, Blaydes, Grimmer, and McQueen (2018) have shown how text-as-data methods can augment and strengthen historical analyses of theoretical concepts. Political theory has also benefited from text-as-data work that helped to establish the authorship of disputed *Federalist Papers* (Mosteller and Wallace 1964/2008) as well as attributing certain anonymous works to Thomas Hobbes (Reynolds and Saxonhouse 1995). Related disciplines like history, classics, law, and literary studies have found text-as-data approaches to be similarly helpful as a means of tracing topics and themes (Rhody 2012; Mimno 2012; Jockers 2013).

Such work, however, only begins to demonstrate the potential utility of text-as-data methods for political theorists and those engaged in conceptual research. New methods of computationally analyzing large collections of text using word vectors can go beyond topics or authorship, directly revealing the changing meanings of individual words and concepts across time. Concepts like rights, equality, peace, and democracy are both important to understand on their own, and also function as key variables in many scholarly hypotheses. Word vector methods that track the changing meaning of such concepts will have broad utility for political scientists.

Consider, for instance, the idea of equality in America. The meaning of equality in America has been central to our political culture; as Alexis de Tocqueville put it in *Democracy in America*, "the gradual development of the principle of Equality" underlies all other political dynamics and forces in the United States. Yet our contemporary vantage point tends to flatten our view of the past, making it hard to read the potentially alien meanings of equality in, say, the Founding era or the Reconstruction era on their own terms. In the current moment, we tend to strongly associate the idea of equality with progressive struggles for justice on behalf of marginalized groups. Computational methods allow us to step outside of our moment, enabling us to understand the meanings of concepts in the divergent cultural contexts of the past. Our position in the present moment, for instance, makes it surprising to learn that the most vocal advocate of equality in the era between the first and second world wars was Germany's Adolf Hitler. Hardly viewed by history as a model egalitarian, Hitler was nevertheless extremely fond of invoking equality in his speeches and quotes to the press, arguing in January of 1937 that Germany had finally won the "battle

for equality" with other European powers that he had spent years urging as Germany's national mission. The use of equality language among nations, in matters of norms, peace, standing, and war, also dominates American discourse on equality in the interwar period. In the same period, equality discourse around women and African Americans is comparatively rare, complicating the historical story about the close association of equality with marginalized identities like race or gender. As I show in §7, word vectors can also help complicate our view of equality in the present.

Word vectors are a relatively recent methodological innovation that allows us to unearth and track such unexpected shifts in a word's meaning. These new computational methods can potentially allow analysts to excavate the historical layers of the concept, offering a time-sensitive genealogy of its meaning to supplement their own subject area expertise.

3 What are Word Vectors?

Basic computational linguistics can provide interesting information about a word in a given set of documents (known as a corpus). We can, for instance, count the number of times that a given word is used in each document or the percentage of documents containing the word in any given year. In an archive of newspaper articles, we can see where the word appears (the headline, the abstract, the first paragraph, etc.). To understand what the word means when it is used, however, requires not merely counting the uses of the word but somehow quantifying the meaning of the word (Kulkarni et al. 2015). Word vectors attempt to do this. In this section, I offer an intuitive, minimally mathematical discussion of the general theory and mechanics of word vectorization. Once we have a sense of how such models work, we can then turn our attention to the acute dilemmas they present for temporal questions and then to implementations for small data sets that provide possible solutions.

3.1 Distributional Semantics

Word vectorization methods are an implementation of a much older linguistic theory known as the distributional hypothesis. Simply put, the distributional hypothesis argues that the meaning of a word can be extracted by looking, over many texts, at the words that occur around it. As linguist J. R. Firth emphatically put it, "You shall know a word by the company it keeps!" (Firth, 1957, 11). The distributional hypothesis argues that, independent of any other context or even grammatical order, a systematic collection of word collocations can allows us to make semantic 'sense' out of words. In other words, linguists have argued that the words that appear near one another contain a surprising amount of information about the meaning of a given word and that a "difference of meaning correlates with difference of distribution" of these collocations (Harris, 1954,

156). Words that appear in similar contexts of nearby words have similar meanings.

The power of the distributional hypothesis lies in its generalizability. It can be applied to any language, or even to individual corpora within a given language; it requires no prior input of dictionaries or grammatical structures. It deduces – and here we should probably say 'we deduce,' since there is evidence that this is how humans learn language – the meanings of words based on considering the collocations of all words that occur in a given corpus of texts. The distributional hypothesis is the backbone assumption that drives word vectorization models (Turney and Pantel 2010).

3.2 Turning Words Into Vectors

How might we approximate the distributional semantics model of human language acquisition to teach it to a computer? We might begin by looking at simple word co-occurrence, a very basic way of expressing the understanding that words that occur in similar context have similar meanings. To look at the co-occurrence of words in a given text, we could take every possible pair of words in a text, and then calculate the conditional probability $P(a|b)$ of word a occurring within, say, five words of word b in the text. Since different words have different independent probabilities of occurring, though – some words are more common than others – we want to calculate what is called the point-wise mutual information (PMI) of the pair: in essence, how much more are we seeing a given pair of words occurring together than we would expect to at random, given the independent probability of seeing each word. PMI values can be positive or negative, but $PMI(a, b) = 0$ when words a and b are independent, i.e. when the presence of one word has no impact on the likelihood of seeing the other. The calculation of the PMI of a given pair of words is then approximated using linear algebra:

$$PMI(a, b) = \vec{\mathbf{A}} \cdot \vec{\mathbf{B}}$$

where the model makes the assumption that PMI can be approximated as a single (scalar) product of two equivalent-length vectors, $\mathbf{A}$ and $\mathbf{B}$.

But what are these new vectors that represent each word? Thinking about words a and b as vectors allows us to conceptualize something important. If we are interested in meaning, we are probably more interested in synonyms than in mere collocation: that is, we want to know which words appear in similar contexts, and such words are thus not likely to appear together. Hence, we are interested in words a and b that occur with the same frequency relative to all other words w. The mere co-occurrence of a and b is far less useful than the comparison of both a and b's co-occurrence with w. In other words, we are more interested in comparing the meaning of the two words, a meaning that we can deduce from the context in which each word appears. In broad

conceptual strokes, in a case where words a and b are functionally equivalent – exact synonyms or antonyms, for instance, always used in the same contexts – their vectors will contain the same values along all dimensions, yielding the highest possible PMI value. Words a and b that are unrelated to each other – whose contexts of appearance in the text are fully independent from one another – will have orthogonal vectors, whose product will yield a PMI of 0. Words a and b that are inversely related – less likely than chance to appear in the similar contexts – will yield negative PMIs.

But what, exactly, are these vectors of values? If each word is represented as a vector of values of equivalent dimensionality n, there is an n-dimensional *reference vector* which we can imagine as being a vector of points upon which each word is scored or compared. Let's imagine we want to compare words a and b. We could, for instance, imagine a vector for word a that has as many dimensions as there are unique words w in the text, and scores the co-occurrence (how often they appear together within a fixed window around each word) of word a with each other word w in the text. The reference vector, then, would be a vector of w-dimensionality of all unique context words w. We would then produce another vector (again with w-dimensionality) of co-occurrences of all context words w with a second word b. We would then be able to compare $\vec{\mathbf{B}}$ to $\vec{\mathbf{A}}$ by taking their scalar product, which would give a single number that we might term a similarity score.

One thing that we might notice, however, is that every context word w is not equally useful in helping us to distinguish the difference or similarity of words a and b. Some words will be louder or brighter sources of information; many words mean similar things and can be clustered together without a loss of information. In other words, rather than a w-dimensional reference vector, we can actually describe word a (and every other word in the text) with a much, much smaller n-dimensional reference vector against which a is scored. The scores of word a against each element of the reference vector, when all taken together, comes to represent the abstract meaning of a. One could think about the reduction from w to n dimensions as a reduction from the vocabulary of all words in the text to a much smaller vocabulary of concepts: for example, from rating the dog-ness, fox-ness, squirrel-ness, etc. of a word to merely its animal-ness. (Though this example might aid our conceptualization of dimension reduction, it is important to note here that word vector dimensions don't actually refer to anything directly like concrete concepts.) These values of the vector of a given word, taken together, provide its semantic *address* in a given corpora's universe.

3.3 Algorithms and Implementations

Reducing the w-dimensional vector to smaller n-dimensionality is a form of unsupervised learning; in other words, it does not require a human to specify a set of concepts against which words are

scored. Like unsupervised topic modeling, it takes a user-specified value of n (between 100 and 500 dimensions is common) and generates n points against which each word is scored. There are several popular methods by which this functional compression to lower dimensionality takes place, some of which also modify some of the details of the preliminary steps conceptually outlined above. In general, contemporary methods of word vectorization rely on a neural network approach to translate a large, sparse matrix of values into a dense, low-dimensional vector space via a hidden layer of weights.

The details of neural networks and the algorithms that train them are beyond the scope of this paper; for our purposes, we can imagine a neural network as an extremely complex function that takes a huge number of inputs (a corpus of words and each of their surrounding window of context words) and produces a huge number of outputs (vectors of n-dimensionality for each word).[7] The word vectors produced by the model are actually one of the hidden layers of weights in the neural network; the actual final outputs of the neural network are predictions about words that co-occur.

The most popular implementation of word vectors, `word2vec`, structures this network in two different ways. In the case of the skip-gram with negative sampling (SGNS) algorithm that this paper utilizes, the model trains itself on the corpus by working to improve the accuracy of its predictions of words it knows co-occur in the corpus and minimize the likelihood of "negative samples" of words it knows don't co-occur (Mikolov, Yih and Zweig 2013). The model seeks to give each word in the corpus values (weights) along each dimension of the n-dimensional vector that minimizes the cost function: i.e. that reduces the number of incorrect predictions about word co-occurrence. The weights of the model are typically initialized randomly, and then the model works to tweak the weights and biases at every term in the function until it reaches a local maxima in the likelihood of predicting a set of context words given a single input. This training of the network is done through stochastic gradient descent and backpropagation. The final weights of the model correspond to the values of the word vector for each word.[8] By contrast, the continuous bag-of-words (CBOW) `word2vec` architecture essentially inverts the SGNS model: it trains a neural network to optimize its prediction of words given the context around them, rather than trying to predict the context given a single word. As the developers of these models put it, "the CBOW architecture predicts the current word based on the context, and the skip-gram predicts surrounding words given the current word" (Mikolov, Chen, Corrado and Dean, 2013, 5). SGNS tends to run more slowly, but to be better for infrequent words and for semantic tasks, which

[7] For the modeling specifications and computational details behind the `word2vec` implementation used in this paper, see Mikolov et al.'s original papers (Mikolov, Yih and Zweig 2013, Mikolov, Chen, Corrado and Dean 2013, and Mikolov, Sutskever, Chen, Corrado and Dean 2013) as well as several useful explanatory notes elaborating on `word2vec` (Goldberg and Levy 2014, Rong 2014). Excellent general online introductions to neural networks include the four video series produced by Grant Sanderson (www.3blue1brown.com) as well as the eBook introduction to the topic at http://neuralnetworksanddeeplearning.com/

[8] For a more detailed discussion of the SGNS algorithm, see Goldberg and Levy 2014.

makes it the architecture of choice for smaller corpora (Mikolov, Chen, Corrado and Dean 2013).

Using either the CBOW or SGNS architecture, the neural network is trained on a given corpus of interest, producing a vector of weights associated with each word in the context of that corpus. Several straightforward implementations of these methods exist in both the R and Python programming languages. In R, the package `wordVectors` builds on `word2vec`, while in Python, the `gensim` package will allow you to build SGNS or CBOW models with greater freedom to set the hyperparameters and structure the analysis; the `gensim` implementation is recommended for temporal analyses and is used here.

3.4 Training & Corpus Size

Using these implementations, one can either run analyses on existing, trained word vectors or one can train new word vectors on a given corpus of interest.[9] While most existing trained `word2vec` embeddings were generated from large corpora with millions or billions of tokens, there is no hard minimum requirement on the volume of texts required to train word vectors.[10] Though the conventional wisdom is that larger corpora will train higher quality vectors, several promising lines of research are opening doors to smaller corpora.

First, small corpora word vector analyses can benefit from the use of transfer learning, also known as pre-training or fine-tuning, where the model has non-random initializations. Most prominent in computer vision studies that attempt to train image classifiers, transfer learning has shown that gains in performance and efficiency are possible when a machine learning model is initialized with weights from a previous model, rather than starting the learning process from scratch (Pan and Yang 2010). Translating this insight to word vector models, one might use vectors from an entire corpus to initialize the first slice in a diachronic analysis, as I discuss in the next section, or one might use a small corpus of interest to fine-tune vectors trained on an entirely different, more universal corpus (Howard and Ruder 2018).

Another development in small corpora word vector analysis relies on statistical insights around bootstrapping. One concern about small corpora is that the vectors they produce are highly sensitive to single documents, corpus size, and document length, generating divergent model outputs based on small changes to these corpus characteristics (Antoniak and Mimno 2018). Antoniak and Mimno have shown, however, that averaging over the model outputs from multiple bootstrap

[9] The public release of `word2vec` contains pre-trained vectors, as well as links to a number of other pre-trained vectors (https://code.google.com/archive/p/word2vec/). Several diachronic studies have also publicly released their trained vectors, including Hamilton, Leskovec and Jurafsky 2016*b*.

[10] While the quality of embeddings, in general, will increase as corpus size increases, high quality word vectors have been generated from modest corpora. The demo_words/text8 corpus released with `word2vec`, for instance, is used to introduce users to the utility of word vectors and contains only about 250,000 unique words. Even toy examples of training `word2vec` on a handful of sentences have been shown to retrieve semantically revealing embeddings. In other words, both the quality and quantity of the corpus matter, and as with all unsupervised techniques, validation is more important that arbitrary benchmarks for corpus size.

samples of documents can produce stable, reliable results from small corpora.

3.5 Uses for Word Vectors

Once a corpus has been vectorized, each word essentially has an address or assigned location in n-dimensional space. Just as (x, y) coordinates allow us to locate a position in two-dimensional space and can be added or subtracted to move around the grid, word vectors locate words in a semantic space of n-dimensions, and can be added or subtracted to move around the space. Hence, the analogy *king is to man as woman is to what?* is a logical expression, but embedding these words in vector space makes it an algebraic expression, not just a logical one. As noted at the outset of this paper, by starting with the vector for 'king,' subtracting the vector for 'man,' and adding the vector for 'woman,' the word located closest to the position you end up is 'queen.' Word vectorization models trained on large corpora have performed astonishingly well on these kinds of language analogy tasks. They are very good at identifying the fourth word in an analogy, or at identifying the word in a cluster of four that is least like the others.

Word vectors can also be directly compared to one another to produce a similarity score between words. The most common method of doing this is by cosine similarity, where $\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$ gives the cosine similarity, ranging between -1 and 1, of two vectors $\mathbf{A}$ and $\mathbf{B}$, drawn from the origin. The semantic similarity between two words is approximated by their cosine similarity, with values closer to 1 indicating greater similarity (Turney and Pantel 2010; Antoniak and Mimno 2018).

Another way of thinking about this is that we can semantically map a given word, like "equality," by looking at its cosine similarity scores with other words (e.g. "race" or "gender"). The changing proximity of two words will reflect shifts in the cultural meaning of a given word: for instance, the proximity of "woman" and "homemaker" over time will reveal important semantic information about the cultural meaning of the word "woman." Similarly, when people think about equality in any given era, how close is gender or race in their thoughts? Or what about the proximity of equality to words like liberty or dignity? Cosine similarity scores (either for pairs of words, or lists of closest words like in Hamilton, Leskovec and Jurafsky 2016*b*) can allow analysts to track these changing semantic relationships, which in turn reflect a picture of the thick, cultural meaning of equality as it changes over time.

4 The Problem of Time

The word vectors produced from a corpus express the language regularities and semantic relationships in a given corpus. Another way to say this is that the corpus is "the central object of study" for word vector methods and the goal is to understand and explore the distinct semantic universe

of the corpus and, by extension, the world that produced the corpus (Antoniak and Mimno, 2018, 107). For example, analyses on distinct corpora of texts can reveal what we might term ideological biases in each corpus. One `word2vec` analysis of articles from left-leaning, centrist, and right-leaning news sites done by ProPublica found that there were marked differences in the words with the closest cosine similarity scores to "Clinton" or "abortion" in each corpus of news articles (Larson, Angwin and Parris 2016). Other word vectorization analyses have revealed that corpora reproduce gendered or racial biases and stereotypes (Bolukbasi et al. 2016; Caliskan, Bryson and Narayanan 2017; Garg et al. 2018).

Word vectorization allows us to map the semantic universe of a given corpus, and then to compare universes to each other in order to track semantic differences. These universes might be different subsets of the world at the given moment in time – like the example of different sets of media with certain ideological orientations – or they might be the same subset of the world at different moments in time. For instance, we can imagine that a sample of articles taken from the *New York Times* in 1870 will represent a distinct semantic universe from a sample of *Times* articles taken in 1970. This section explores the challenges associated with using `word2vec` to produce valid results for this type of chronological comparison.[11]

4.1 Challenges for Diachronic Analysis

As with all methods for computationally analyzing text as data, `word2vec` users need to be aware of how pre-processing and hyperparameter settings can influence the outcomes of their models.[12] Word vectorization analyses conducted over time, however, face three additional validity challenges which have been either generally under-appreciated in previous studies or have not been addressed with an eye to valid semantic analyses in small-scale corpora. All of three of these problems emerge from the norm of slicing a single corpus into time intervals in order to model each semantic universe separately, producing word vector relationships for each universe, and then tracing the changes in those semantic relationships across time.[13]

[11] Although testing possible solutions is beyond the scope of this paper, many of the challenges discussed in this section also appear to apply to out of corpus cross-comparisons in a single time period, particularly concerns about spatial non-comparability and language stabilization.

[12] For analysts who want to use `word2vec` to confidently assert something about the semantic universe represented by a single given corpus, several types of unique stability challenges exist. First, the embedding algorithms are sensitive to even seemingly minor variations in the size of the corpus and its component documents, the presence or absence of single documents, and the random seeds; previous studies have found that such challenges can be addressed by averaging the results over multiple bootstrapped samples of the corpus and supplying confidence intervals around values of interest like cosine similarities between words (Antoniak and Mimno 2018). (There are other ways of approaching the problem of statistical significance for model outputs (see Han et al. 2018) but bootstrapping provides programmatic simplicity and reproducability with small data sets.) Second, results shift with variations in the user-specified hyperparameters of the selected algorithm, like the dimensionality of the vectors, smoothing, context-windows, and sample sizes, suggesting that analysts should select and tune their algorithms by testing for successful task performance (Levy, Goldberg and Dagan 2015).

[13] As noted already, the vanguard of word vectorization is in dynamic models which do not appear to require such slicing. Several very recent computer science conference papers (Bamler and Mandt 2017; Yao et al. 2018) show promise on this front, but such implementations are not tested here. *See* fn. 4, above.

Arbitrary Cut-Points

First, slicing a corpus into time eras introduces arbitrary cut-points between eras. The selection of these cut-points – both their location and number – can have an effect on the embeddings produced and thus on the similarity scores. While the size of the eras will be determined to some extent by the research question and by the size and scope of corpus, selecting different era sizes has known effects on trends and the smallest feasible era size should be utilized to avoid losing trend information (Box-Steffensmeier et al. 2014). In addition, semantic shifts can be produced quickly by dramatic single day historical events like September 11, 2001 or October 29, 1929, or by less visible but still rapid processes of cultural and linguistic change. Cutting a corpus on one side or another of such events affects model results. While some of these shifts are known and can be accounted for, others are discovered by the model itself and so cannot be accounted for in advance. Thus, any given cut-point introduces unknown and arbitrary effects into the trends the model produces.

Language Instability

The second problem that analysts of semantic changes over time face is issues of homonymy, polysemy, and instability in corpus vocabulary (Huang et al. 2012). The same concept is often referred to using different words over time, limiting the ability to use cosine similarity between a given pair of words over time as synonyms shift in and out of fashion. For example, if we want to track the relationship between equality discourse and black Americans, we will need to choose a term consistent in meaning over time and distinguishable from other meanings of the term. Our challenge becomes clear when when we recognize that today "African American" is used to describe Americans of African descent, but this term only came into use in the last three or four decades; prior popular terms used in print for the same set of people have included "Negro," "Black," "Colored," and (if you go back far enough) "Freedman" (plus all respective pluralizations). The polysemous character of the word "black" – both a race referent and a general color – increases the difficulty. The underlying concept "Black American" may exist across time, but both its cultural meaning and the specific word used for the concept will have shifted.[14] To produce valid measures of its relationship with equality across time, then, one has to account for language instabilities like these.

[14]In a minority of cases, the word itself may not continue to exist across time, even though an underlying concept does endure. The African American example illustrates this disjoint between the concept of a "black race" and the dramatically changing words used across time to signify that concept. At the same time that the synonyms are shifting, however, the cultural meaning of the underlying concept is also shifting – and it is this later shift that the word vector analyst seeks to track. In other words, to attempt to stabilize a concept down from a collection of synonyms to a single word is necessary in some cases to allow us to study the concept at all. When I use "concepts" subsequently in this paper, it is this idea to which I am referring.

Spatial Non-Comparability

Finally, word vectorization models suffer from a particular and (at least among non-computer science practitioners) under-appreciated challenge of what I will term spatial non-comparability. Recall that our model produces word vectors with length n, where each element can be understood as a coordinate that locates or embeds a given word in n-dimensional space. Words located closer to one another in this space are more similar in meaning; cosine similarity between two vectors gives a kind of proximity score. The overall space is defined by a set of n basis vectors, which allows the analyst to orient themselves within the space and meaningfully compare vectors and distances. The simplest example of this is in the familiar graph of 2-space, where the axes x and y serve as reference vectors that allow us to plot and compare other vectors of given magnitudes and directions.

A temporal analysis wants to compare cosine similarities (the closeness of vectors) from across different models. But word embeddings produced by stochastic processes like SGNS from different time slices will embed words in non-aligned spaces defined by different basis vectors. This precludes direct comparison of cosine similarity across distinct corpora (Hamilton, Leskovec and Jurafsky 2016*b*). In fact, it even complicates our ability to compare different modeling runs on the same corpus, where nearest neighbors might remain the same but coordinates might shift (Kulkarni et al. 2015). This all makes sense, actually, since the basis of any diachronic semantic analysis is the recognition that the meaning of a word changes over time; correspondingly, we would expect the meanings of most words (and even the list of words) in a given space to shift, and it would be reasonable to expect a corresponding shift in the structure of the space itself.

An example might help to clarify this point. Imagine a graph that plotted per capita consumption of fresh fruit each year in the United States over time. Such a graph would allow confident discussions in trends in fruit consumption over time. Now imagine that instead of each point representing the annual per capita consumption of all fruit, each point represented the annual per capita consumption of some specific fruit, where the specific fruit used changed each year. One year, we would plot the per capita consumption of bananas, the next year mangoes, the following year apples. Such a graph would be in some unknown way related to overall fruit consumption trends, but would much more prominently be related to the idiosyncrasies of the consumption of a given individual fruit in a given year. Without a way to convert individual fruit consumption to a common, comparable baseline, it would be challenging to speak confidently about fruit consumption trends from this graph. While the analogy to spatial non-comparability is inexact, this example gives some sense of the alignment problem faced by trying to compare values with different baselines.[15]

[15]Another example that might clarify the alignment problem is its similarity to the problem faced in a factor analysis run on similar but non-identical datasets, where analyses might produce similar factors but mapped in a

4.2 Analyzing Vectors Across Time

While the first two of these problems are common to all efforts to analyze semantic shifts over time, addressing them within the context of word vectorization presents particular challenges. The third problem is unique to `word2vec`. There are many possible ways to implement time-sensitive word vectorization that attempt to address these issues; I describe four in the subsection that follows. I then implement tests of each using a small, relatively sparse dataset typical of corpora used by social scientists and scholars in the humanities. Rather than work out a theoretical solution to these problems, I create a set of gold standard semantic tests, and empirically determine which implementation allows `word2vec` to overcome these challenges and excavate known trends in the corpus.

The simplest method of conducting an analysis over time using `word2vec` is to cut the corpus into chronological time slices (days, months, years, etc.) and separately model each slice. We will call this a **naive time series model**. This method produces a set of word vectors for each interval of time, from which cosine similarity scores between pairs of words can be computed. Bootstrap re-sampling of the data and repeated modeling of each slice allows for the generation of standard errors and confidence intervals around such scores; these scores can then be compared across the slices to measure semantic changes across time. The cosine similarities for terms of interest can be compared (e.g. comparing the cosine similarity score of "race" and "equality") at slices t and $t+1$ to ascertain, with some level of confidence, if the meaning of the word of interest (here, "equality") is moving. The naive time series assumes that there are no problems of spatial non-comparability across time slices and does not account for the effect of arbitrary cut points; it serves as a functional baseline or null result.

The **overlapping time series model** attempts to smooth out differences across time slices by cutting some of the edge texts from slice t into the corpus at slice $t+1$, and so on. Otherwise, the same methods are applied and the same results are generated as from the naive time series. This method decreases the chance of spurious results as a consequence of arbitrary cut-points between time slices, but does not directly address problems of spatial non-comparability across slices.

Rather than initialize the `word2vec` model for each time slice with random weights, the **chronologically trained model** utilizes the word vectors from $t-1$ to initialize the model for slice t (Kim et al. 2014). This method assumes, essentially, that word meanings and relationships in slice t begin semantically where $t-1$ ended. The first time slice, t_1, is initialized with the vectors from the full corpus. This method utilizes the leverage offered by pre-trained vectors and provides some semantic linkages across slices, but does not directly address spatial non-comparability. The training on t and production of results then proceeds as in the naive time analysis.

different order.

Finally, the **aligned time series method** generates vectors for each slice as in naive time analysis, but then seeks to address the spatial non-comparability of slices in a post-modeling alignment phase (Kulkarni et al. 2015; Hamilton, Leskovec and Jurafsky 2016*b*). One approach to this task is an orthogonal Procrustes matrix alignment. The alignment process requires that the analyst choose one anchor slice in the corpus, to which all other slices are aligned.[16] At the alignment phase, vocabulary is necessarily limited to words present in all time slices, limiting the information available in any single slice and potentially limiting the utility of this method for corpora which extend over long periods of time or where vocabularies change dramatically.[17] This type of alignment model has been used with some success to align embeddings across languages, on very large corpora.[18]

The gap between the tests previously done on some of these methods (produced in computational linguistics or computer science, with extremely large data sets) and typical use cases for social scientists and humanists is profound. For example, the aligned time series analysis performed by Hamilton et al. (2016*b*) utilized several data sets, including the Google N-Gram data set, which is constructed from 6% of all books ever published and contains $8.5 * 10^{11}$ tokens (roughly, words). Their smallest corpus, a genre-balanced collection of literature, contains 410 million tokens. While some social scientists and humanists employ comparatively large data sets, time analysis word vectorization methods that will be broadly useful to practitioners must apply effectively to substantially smaller corpora. Existing diachronic analysis papers provide little guidance on this front. Diachronic word vector methods will only be useful insofar as practitioners with smaller corpora can feel confident that the models are retrieving valid semantic information about trends in the corpus. In the next sections, I test the relative efficacy of these four models on just such a semantic task: recovering known semantic relationships in a previously coded, relatively small corpus. I compare the performance of these four models against one another on the same corpus and same task, generating empirically validated methodological recommendations for practitioners.

[16] Yao et al. 2018 test various alignment methods. They test both an orthogonal transformation (solved with an n-dimensional Procrustes alignment) as well as the linear transformation used by Kulkarni et al. 2015, which involves aligning locally by solving an n-dimensional least squares problem of k nearest neighbor words. In their tests, they find that Procrustes "performs well, as it also applies alignment between adjacent time slices for all words. However, [linear transformation] does not perform as well as others, suggesting that aligning locally (only a few words) is not sufficient for high alignment quality" (Yao et al., 2018, 7). For these reasons, Procrustes was chosen as the alignment method in this project.

[17] This model requires that the same set of vocabulary be present in each slice, to allow alignment of the matrices. For very large corpora, this is a merely footnote to the model, but in small corpora – because of the possible variation in vocabularies – this is potentially a severe restriction to analysis.

[18] See, for instance, Facebook Research's MUSE (Multilingual Unsupervised or Supervised word Embeddings) Project at https://github.com/facebookresearch/MUSE.

5 Data and Methodology

This project seeks to validate best practices for using `word2vec` on diachronic questions with smaller corpora. In this section, I begin by discussing the challenges of validating diachronic word vector models, describing typical validation methods and justifying my choice of gold standard semantic test. I then describe my corpus of texts and how I modeled them using human coding and supervised topic modeling to produce the gold standard description of the semantic relationships in the corpus. I follow this with the details and implementation of each word vectorization method. I end this section by spending some time describing the assumptions and pre-processing choices that allow me, in this specific case, to use supervised topic model results to empirically validate unsupervised word vector models. The results of the tests of each word vector model against the gold standard, and the general findings, are described in Section 6.

5.1 Problems of Validation

Word embedding models are typically assessed on language tasks that mirror how well the model "understands" the underlying language structure of the corpus. Computational linguists have relied on sets of grammatical, knowledge-based, and semantic analogy tasks, (Mikolov, Yih and Zweig 2013; Pennington, Socher and Manning 2014; Levy, Goldberg and Dagan 2015) as well as word similarity tasks (Hamilton, Leskovec and Jurafsky 2016*b*; Bruni et al. 2012; Levy, Goldberg and Dagan 2015). In these assessments, differently specified models are scored and compared based on how accurately they are able to answer questions like:

- Athens is to Greece as Rome is to _____?
- Mexico is to peso as USA is to _____?
- Dad is to mom as father is to _____?
- Running is to ran as knowing is to _____?

The answers to these questions are produced using vector arithmetic and cosine similarity scores, where the model answers with the word closest to the position in space produced through vector addition and subtraction of the preceding three elements. Such tasks assess how well the model can grasp the language structure of the corpus and answer questions with known answers; models assessed in this way are typically trained on exceptionally large corpora.

Similarly, some large scale diachronic models have been assessed with analogy tasks that rely on extrinsic knowledge that changes in each time slice: for instance, the ability of the model to correctly associate the last names of all United States presidents with their terms or South African political leaders and their parties (Yao et al. 2018; Arnold et al. 2018). Other diachronic studies,

however, have lacked an extrinsic method of evaluation (Kim et al. 2014) or have relied on human-created semantic tasks intrinsic to the substantive questions of interest (Hamilton, Leskovec and Jurafsky 2016*b*; Kulkarni et al. 2015; Yao et al. 2018).[19] Similarly, as I describe below, this paper departs from validation tasks like analogies and word similarities. Instead, I construct a gold standard semantic task specific to the substantive research question in order to validate the word vector models.[20] In general, the diachronic literature agrees that assessing the fit of the models comes down to how well the models discover known semantic relationships between words; in other words, "a good embedding provides vector representations of words such that the relationship between two vectors mirrors the linguistic relationship between the two words" and the best way to assess whether this has occurred is research question dependent (Schnabel et al., 2015, 298).

5.2 Data

For this project, I constructed a corpus of newspaper articles ($n = 3,105$) from the *New York Times* (NYT), *Reuters*, and the *Associated Press*, accessed via the NYT Articles API.[21] All articles from 1855 to 2016 with the word "equality" in the headline were downloaded. Headline restricted articles were chosen in order to construct a corpus of articles centrally, rather than incidentally or tangentially, concerned with equality. The headlines, first paragraphs, and abstracts from each article were used for analysis.[22] The corpus was divided into seven 25-year time slices; the proportion of articles in each time slice that are centrally about equality ranges between .6 and 3.4 articles per 10,000.[23] Due to both this varying newsworthiness of equality and the general increase in the volume of news articles produced in later slices, there is significant variation in document counts across slices. In chronological order by slice, there are 80, 102, 496, 1137, 660, 259, and 371 documents, respectively.

[19]Existing diachronic studies may employ this tactic for several reasons. In the case where learning the set of meanings in a given, smaller slice of the corpus is the task of the model, grammatical or knowledge based validation tasks are less useful because it would be unclear to the analyst whether lower success rates on analogy tasks are the result of model problems or (potentially interesting) semantic differences in a given slice or corpus of texts. The analyst has no reason to expect certain analogies, grammars, or word similarities to remain equally salient or discoverable over time: as the meaning of words and emphasis in language changes, so too would performance on a static set of language tasks. Moreover, a given time slice may contain little overlap between, for instance, discussions of presidents and – as is the case in this paper – discussions of equality. This is a semantic feature of the corpus, not a modeling bug. It follows from this that there is little reason to believe that model success with knowledge or grammatical analogies corresponds to model success tracing diachronic semantic relationships. The human-created semantic tasks that this paper and others rely on, then, appear to be a better fit for the specific validation challenges of diachronic word vector models.

[20]This method of validation was first suggested to me by Yao et al. 2018, who quantitatively assess their model outputs by comparing them to ground truth semantic topic codes, which are assigned by the *New York Times* newsroom to the newspaper articles in their corpus. Though the pre-existing topics assigned by the Times did not fit this project's research question, it did suggest the possibility of constructing a question-specific topical gold standard against which word vector models could be validated.

[21]For replication data and code, see Rodman 2019.

[22]The New York Times API does not allow users to download the full text of articles; abstracts or first paragraphs were not available for all articles. After concatenating all available text data for each article – the headlines, abstracts, and first paragraphs – the average length of an individual document in the corpus is 66 words. This is the same number of words as the text in this footnote.

[23]There were only 12 years in the last time slice (2005-2016, inclusive).

5.3 Producing the Gold Standard

To produce the gold standard against which the word vector models were validated, I first created a codebook for my newspaper corpus based on spot reading and computational exploration. The topics in the codebook emerged organically from an exploration of the corpus (Saldana 2009).[24] These topics were intended to capture who, in a given article, is seeking, achieving, needing, or being denied equality – in other words, which group was associated with the equality discussed in a given article. This codebook was refined iteratively and finalized during several preliminary rounds of trial coding with two undergraduate coders. The final codebook contained fifteen mutually exclusive and exhaustive topic codes, including a "misc." topic.[25]

Next, using this codebook, the coders each coded the same simple random sample of 400 articles from the corpus, with an overall inter-coder agreement of 89%. After every 100 articles, coding disagreements were referred to me and resolved in conference with the coders; I also spot checked their coding on articles where they were in agreement. This coded collection of 400 articles served as the training set for the supervised topic model.

This training set was then used to run a supervised topic model on the full corpus of all articles using the R package `ReadMe`, which utilizes a training set of documents hand-coded with exhaustive and mutually exclusive topics to compute the proportion of documents in each topic in a second test set (Hopkins and King 2010). Following Hopkins and King's implementation of a proportions-over-time analysis, I split the test set into 25-year eras and used the full training set to model each era separately.[26] Bootstrapping ($n = 300$) was used on each era to produce means for the document proportions in each topic. The shift in the proportion of documents in each of the `ReadMe` model's topics matches known historical shifts in the use of equality language – we see, for instance, a large spike in gender articles in the suffrage era of 1905-1930; a similar spike is present in both international relations and German focused articles in the era leading up to and including World War II – which provides overall inductive validity for the model.

[24] In addition to spot reading, I performed exploratory unsupervised topic modeling on the corpus. Such methods have been shown to effectively parse topics in newspaper corpora (Newman and Block 2006; Yang, Torget and Mihalcea 2011). The purpose of this preliminary modeling was to give me a sense of what topics might exist in the corpus. I used this to guide my construction of an initial codebook. Each era was fitted with a latent Dirichlet allocation (LDA) model with the VEM algorithm, which assumes that each article is a mix of topics (Blei, Ng and Jordan 2003). Topic models were fitted separately to each era with various user-specified values of k (the number of topics), and then I substantively evaluated the resulting topic lists for semantic validity by looking for distinctiveness between topics and internal consistency within topics (see Quinn et al. 2010). Prior to topic modeling, capitalization, punctuation, symbols, spaces, and stopwords were removed using the R package `quanteda`. This package was also used to stem the unigrams – reducing word-list complexity by aggregating families of words – using the Porter stemming algorithm. I utilized a bag-of-words approach to these elements, which is standard in computational text analysis, breaking each text down into unigrams without reference to word order (Jurafsky and Martin 2009; Hopkins and King 2010; Grimmer and Stewart 2013). Topics were modeled using the R package `topicmodels`. Because the unsupervised modeling was intended to be exploratory and suggestive rather than dispositive, I did not address issues of topic instability (Wilkerson and Casas 2017).

[25] The final codebook topics were gender, international relations, Germany, LGBT, general race/ethnicity, African Americans, students, U.S. government and political parties, workers, companies, religious adherents/institutions, intersectional, Jewish, everyone/abstract equality, and other/miscellaneous.

[26] For the earliest era (1855-1880) the gold standard is manual coding of the 80 documents , due to `ReadMe` modeling instabilities.

Finally, I selected five of the fifteen topics, where the topic could be easily approximated by a single word (see section 5.5 below for more detail on this choice).[27] As Figure 1 demonstrates, the trends in proportions over time in these five topics match historical expectations. These trends in the corpus – the shift in document proportions in these topics over time – are the semantic gold standard against which the four word vector models, described next, will be assessed.

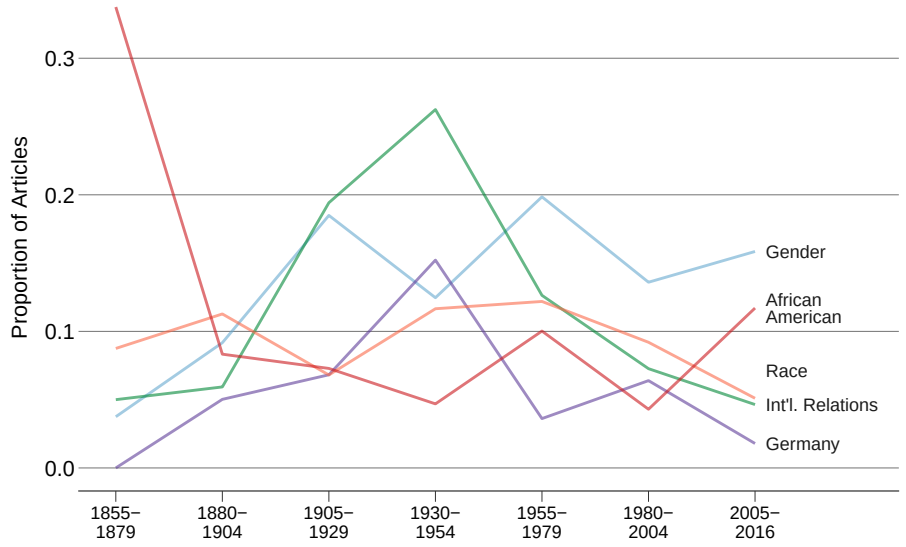

Figure 1: Proportion of documents by topic and era in the gold standard model. (Proportions do not sum to 1 in each era because only five of fifteen topics are plotted.)

5.4 Word Vectorization Modeling

I then produced four diachronic implementations of `word2vec` models. All four of the word vectorization models utilized `word2vec`'s SGNS algorithm as implemented in the Python library `gensim` (Rehurek and Sojka 2010; Mikolov, Yih and Zweig 2013). Except where noted in the implementation of a specific temporal method below, all models follow hyperparameter recommendations from previous literature or `gensim` defaults (Levy, Goldberg and Dagan 2015).[28] All words were converted to lower case and punctuation was removed; the corpus was not stemmed, nor were stop words removed. Each of the seven eras of the corpus was modeled separately.

Two important modeling choices were made, which are recommended for small corpora and diachronic studies. First, each era was modeled repeatedly with bootstrapped samples of the

[27] The remaining eleven topics either could not be approximated with a single word, or their proportions were so small as to make tracking trends over time difficult.

[28] These include: the length of the generated vectors is 100, 200 iterations are made over the corpus, the size of the window of text around each word is 10, and the word frequency threshold is 1. The learning rate is set to 0.025 at the start and linearly decreased to 0.0001.

documents to produce sample means of, and confidence intervals around, model outputs of interest (Antoniak and Mimno 2018). For each bootstrapped sample, n documents were randomly sampled with replacement from the corpus, where n is equal to the number of documents in the era being modeled. As Antoniak and Mimno describe, averaging across many bootstrapped models stabilizes the model outputs (like, in this case, cosine similarity scores) for small corpora, making the analysis less vulnerable to single documents, and allows the computation of confidence intervals around model outputs.

Second, language stabilization and selective stemming was done manually on the corpus prior to modeling to enable tracking of the five test words associated with the five gold standard topics outlined above. This was done to correct for plurals and for the fact that over time synonymous words are used with different frequencies and politically correct language shifts (see Table 1).[29] For instance, the phrase "equality of the races" was quite typical in articles up until the 1955 era; from that point, such language drops out entirely and is replaced by "racial equality." In such a case, we need some way to understand "racial" and "races" as equivalent.

Table 1: Corpus Language Stabilization and Selective Stemming

Topic	Collapsed Word	Constituent Words
Gender	"gender"	Gender(s), Woman, Woman's, Women, Women's, Female, Suffragette(s), Sexes, Sex*
Int'l. Relations	"treaty"	Treaty, Treaties, Pact(s)
Germany	"german"	German(s), Germany
Race	"race"	Race, Races, Racial
African American	"african_american"	Freedman, Freedmen, Blacks, African American(s), Mulatto(es), Negro, Negroes, Colored

To implement each of the four temporal `word2vec` models, the texts for each of the seven time slices were divided into sentences, each of which was then divided into lists of individual words, using the Python package `nltk`. In the naive time model, each corpus of documents was resampled and modeled until the bootstrapped mean around the cosine similarity scores for the five test words stabilized ($n = 100$). In the overlapping time model, the text list T for each era e was sorted chronologically. The first and last 10% of each T_{e_i} were duplicated. These duplicate lists were then appended to the adjacent corpora e_{i-1} and e_{i+1}, respectively. Each era's T_{new} was then bootstrapped as per the naive time model. In the chronologically trained model, the model for T_1 was initialized using the word vectors from a model of the entire corpus. The word vectors from model T_1, when trained, were then used to initialize T_2, and so on through all time slices. Each slice was bootstrapped as per the naive time model. In the aligned model, each corpus

[29] I generated these lists of constitutive terms via a close reading of the corpus and consultation of the Oxford English Dictionary of Synonyms and Antonyms.

slice was modeled separately and the resulting vectors were L2-normalized. The vector spaces in adjacent slices, beginning with e_1 and e_2, were then aligned in sequence by cutting the vocabulary to common words and then using Procrustes matrix alignment (Heuser 2016; Hamilton, Leskovec and Jurafsky 2016*b*). This process was repeated for each bootstrapped sample of each slice; cosine similarity scores were retrieved from slices post-alignment.

5.5 Modeling Assumptions and Data and Pre-processing Choices

Before moving to the findings, it is important to emphasize the limited scope of the validation method of this project. For this project, I produced a unique corpus with specific texts, which allowed me to use the results of a supervised topic model to validate the four word vector models. This was only possible because of the specifics of my corpus and pre-processing steps, and was only desirable because it allowed me to empirically validate general best practices for small corpus diachronic word vector modeling.

It is important to be clear, however: in general, word vector models and topic models are not equivalent. Topic models assign topics to texts, either user-generated topics (supervised topic models) or computer-generated topics (unsupervised topic models). Topics are typically complex ideas expressed through word clouds or collections of key words. Such topic models can be used, as in this project, to track how prominent certain topics are over time by looking at the changing proportion of texts assigned to a topic. `ReadMe`, the method I employ here to assign a topic to each text, does precisely this, taking as inputs a set of human-coded texts assigned to mutually exclusive topics and a set of uncoded texts and producing as output the proportion of all texts in each topic. Word vectorization models, on the other hand, produce a high-dimensional space and embed all the words present in the corpus of texts in that space. These word vector models are unsupervised. Words which are closer to one another in the space are closer in meaning; they have greater semantic proximity. In other words, topic models traffic in the topic(s) of each text, while word vectors traffic in the semantic relationships between individual words in the overall corpus of texts.

To use a supervised topic model to validate my word vector models, I made two important data and pre-processing choices and three key modeling assumptions. The first data choice I made was to limit my corpus to those articles with the word "equality" in the headline. This choice was made because articles with equality in the headline (rather than, say, in a body paragraph) are likely to be centrally, rather than tangentially, concerned with the idea of equality. In other words, I assume that all of the articles in my corpus can be assigned to the topic "equality," in addition to the secondary topic they are assigned by `ReadMe`. The first key assumption, then, is that each article has two topics: "equality" and a secondary topic assigned by `ReadMe` from a codebook of

mutually exclusive options.

The proportion of texts in each of these secondary topics, therefore, corresponds to the centrality of the relationship between that topic and equality in the corpus. If, for instance, only one percent of all the equality articles are about race, it is clear that – at least in this corpus – the equality and race topics are not commonly co-occurring or closely related. If, on the other hand, 65% of the equality articles are about gender, it is clear that the gender and equality topics are frequently co-occurring and closely related: i.e. that there is a closer semantic relationship between the gender and equality topics than between the race and equality topics in this given corpus. The second key assumption, therefore, is that the proportion of texts in each topic tracks the strength of the semantic relationship in the corpus between equality and that topic.

I then selected five topics, from the codebook used to code the `ReadMe` training texts, which could be closely represented by a single word. Recall that topics in topic models are typically expressed using word clouds or lists of keywords. Here, I made my second important data choice, choosing topics (gender, race, African American, Germany, and international relations) which I could represent relatively well with a single word. This required pre-processing the texts fed into the word vector models, performing some language stabilization and selective stemming. After that pre-processing, the third key assumption was made: that the semantic relationship between these individual words ("gender," "race," "african_american," "german," and "treaty") and the word "equality" would mirror the semantic relationship between each of the five topics (gender, race, African American, Germany, and international relations) and the equality topic. In other words, in a highly successful word vector model, the z-score normalized cosine similarity between equality and each of these five words would precisely mirror the z-score normalized document proportions of the corresponding topic from the gold standard supervised topic model.

6 Results

Once I produced the four `word2vec` implementations, I then turned to validation. The models were assessed based how well they replicated the `ReadMe` gold standard of document proportions across five topics: gender, international relations, Germany, race, and African American. The gold standard model produced document proportions across these five topics as depicted in Figure 1 on page 20. The four word vectorization models produced cosine similarity scores for "equality" and the word associated with each topic, as described in Table 1 on page 21. The cosine similarities of those "equality"-word pairs, one score for each era, are shown in the model-by-model plots of word vector outputs in Figure 2.

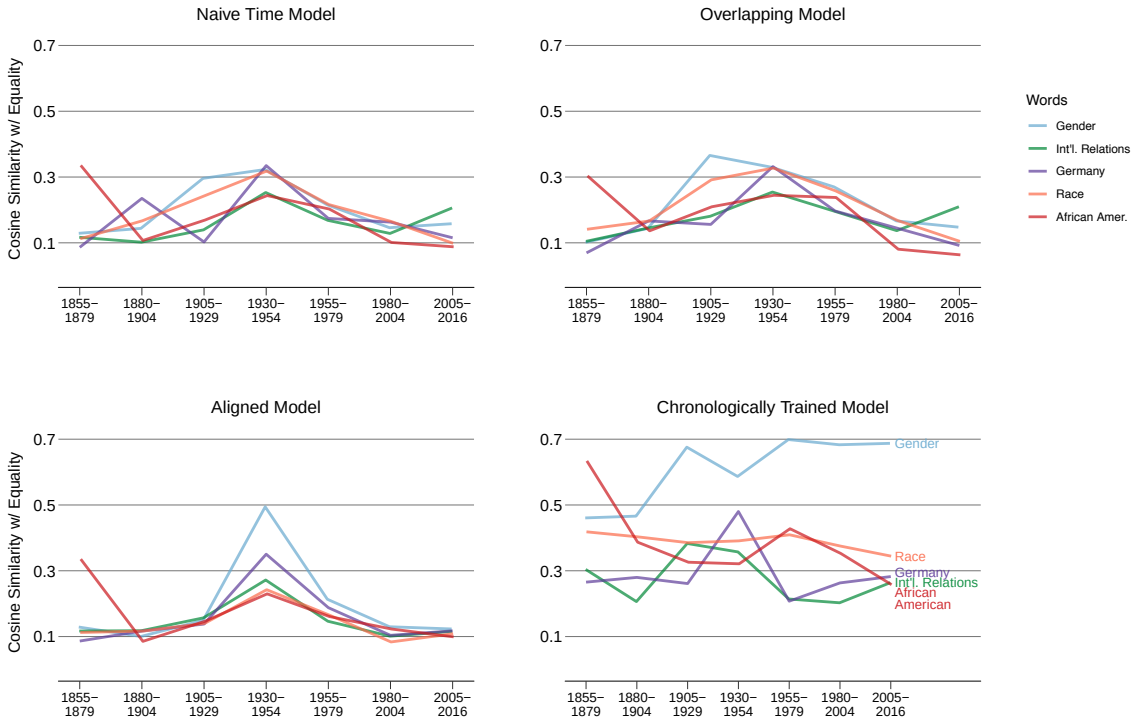

Figure 2: Four Diachronic `word2vec` Model Implementations

6.1 Model Performance

The gold standard was used to assess which of the four `word2vec` models most closely replicated the known corpus trends. After z-score normalizing the gold standard document proportions and the four `word2vec` model outputs, I calculated three fit measures: the sum of the point-by-point deviances between the gold standard and each `word2vec` model, the sum of the point-by-point squared deviances, and the correlations between the gold standard and each of the `word2vec` models (see Table 2 for a summary of how well each `word2vec` model performed on these three metrics).[30]

The normalized outputs offer the same conclusion regardless of the fit measure chosen. Across all three metrics, the chronologically trained model outperformed the other three implementations, more closely positively correlating to the baseline trends and producing the lowest summed point-

[30]Existing studies do little to guide choices for fit measures. Hamilton, Leskovec and Jurafsky 2016*b*, for instance, assess their models using a (for our purposes insufficiently granular) directionality measure: does the model detect the right direction of semantic movement in a small set of known examples (e.g. does the word "gay" move away from the word "happy" and toward the word "homosexual" across the last century)? Yao et al. 2018 – who also use a topical gold standard to validate vector models, similar to this paper – use spherical k-means to cluster their word embeddings. Words "exceptionally numerous" in each *New York Times* assigned topic are taken as ground truth. Then, word co-presence in clusters are then compared to word co-presence in ground truth topics, and the overall accuracy of the model assessed using F_β (the β-weighted harmonic mean of the precision and recall). Such a fit measure, however, requires lists of words most numerous in each topic. `ReadMe` supplies only the proportions of documents in each topic, not labels on specific documents, which precludes the use of a fit measure of accuracy in binary classification.

Table 2: `word2vec` Models Compared to Gold Standard Model

	Deviance	Squared Deviance	Correlation
Naive Time Model	26.612	25.589	0.554
Overlapping Model	25.163	25.601	0.563
Chronologically Trained Model	**22.689**	**21.860**	**0.611**
Aligned Model	25.385	25.775	0.514

by-point deviance and squared deviance from the baseline values. The other three implementations not only failed to as closely reproduce the baseline data, but a one-way ANOVA provides no evidence that the naive, overlapping, and aligned models are statistically distinct from one another ($F(2, 102) = 1.169; p = 0.314$).

The chronologically trained model starts with vectors from the whole corpus, and then is iteratively retrained at each time slice using the vectors from the previous slice. Rather than starting the model with random weights and biases as in the other three models, the chronologically trained model starts with more information in each training cycle. This information appears to allow the model to reproduce the overall structure of semantic shifts in the corpus with higher fidelity, mirroring successes seen with pre-training or transfer learning in other domains of machine learning (Pan and Yang 2010; You et al. 2015). Although this implementation does not directly align the slices to address problems of spatial non-comparability, it appears to functionally stabilize the basis vectors across time slices to some extent.

While the chronologically trained model provides the overall best `word2vec` implementation to track semantic shifts, model performance across certain eras and on certain topics provides additional guidance about the limits of word vectorization methods. As described above, the NYT equality corpus is not large by machine learning standards, and time slices at the beginning and the end of the corpus are particularly sparse. Much of the deviation from the gold standard in the chronological model is centered in those eras; for instance, the 1855 ($n = 80$) and 2005 eras underestimate African American, while gender is overestimated in both eras.

The chronological model also displays limitations on the general international relations topic, roughly reproducing the trend line but not the magnitude of change in that topic (see Figure 3). This is likely an artifact of the limitations of the validation test itself. While other topics are more closely replicated by a single word ("German," "race," etc.), the international relations topic is more of a cluster of concepts, where the test word chosen ("treaty") is likely to capture the rough directionality and shape but not the full magnitude of equality's changing closeness to international relations. As one would expect, this limitation in the ability to reproduce magnitude is most pronounced during the extreme spike in equality rhetoric in international relations in the lead up to, and aftermath of, World War II.

Figure 3: Chronologically Trained `word2vec` Model Compared to Gold Standard

6.2 Recommendations and Best Practices

For practitioners in the social sciences and humanities, word vectors can offer illumination into large corpora, helping to track semantic changes over time. Like all methods, word vectorization is not a substitute for expert knowledge of a substantive topical area; it will complement a thorough exploration of the corpus via spot reading, subject matter familiarity with the context of production, and other computational methods of exploration, like word counts and looking at keywords in context. In other words, as with all unsupervised computational approaches to text as data, external validation of model results is crucial (Grimmer and Stewart 2013).

Backed by this knowledge, word vectorization models can be implemented in ways that increase confidence in the robustness and stability of their results. Bootstrapping as implemented in this

paper is strongly recommended to generate stable cosine similarities and confidence intervals. While this can be computationally intensive, experiments in the literature (Antoniak and Mimno 2018) and in the course of this project show that cosine similarities stabilize at between $n = 25$ and $n = 50$ bootstrapped samples; sparse corpora or time slices would benefit from doubling that number. Based on the outcomes of the tests on this particular project's corpus, analysts should seek time slices with as many texts as possible; depending on document length and how closely each text relates to the words under consideration in the analysis, this analysis suggests that between 100 and 500 documents is a reasonable minimum per slice.

As I have shown, for this data, modeling time slices using a chronologically trained approach best replicates the underlying semantic relationships in the corpus over time. While the generalizability of these results on a variety of data sets was not tested, the literature on transfer learning gives a theoretical basis from which to assume that `word2vec` implementations relying on pre-trained vectors (in this case, pre-trained on the full corpus) will do a better job of modeling semantics in small corpora in general.

Finally, word vector approaches are most effective for studying a single word, rather than analyzing more diffuse notions that might be captured by a topic or word cloud. In other words – as I emphasize in the next section – word vectorization is not a replacement for topic modeling. Once the analyst has identified words of interest, language stabilization needs to take place on the corpus prior to processing. Again, subject matter knowledge and spot reading in the corpus is an important guide to this process. For some words, stemming is all that is required; for others, as in the African American concept in this project, complex political and social processes have dramatically shifted synonyms over time, resulting in the need for more involved pre-processing stabilization steps.

7 Diachronic Analysis in the Field: "Social Equality"

As I have described, the corpus used to test these model implementations was constructed specifically to facilitate comparability between supervised topic model outputs and word vectorization outputs. In this analysis, the corpus was limited to a subset of articles which were explicitly about equality, the analysis was locked into certain topics to compare to equality once the `ReadMe` codebook was finalized, and coders had to be trained and paid to produce gold-standard documents with which to train the `ReadMe` model. These steps were necessary to empirically validate which diachronic implementation performed best on semantic tasks, but do not reflect how analysts will typically use or validate word vector models 'in the field.'

In other words, far fewer limits exist on most `word2vec` analyses. While relatively robust to

scarcity, generally word vector analyses benefit from training on as many corpus documents in each era as are available. Additional word-word pairs can be added to the analysis with relative ease, compared to the expense and inconvenience of manual recoding to add new word-topic pairs to a `ReadMe` model, allowing `word2vec` to be used as an exploratory method. In addition – and most important in terms of its methodological utility – there is far greater nuance available in the word-word analysis of semantics that `word2vec` can perform than was clear in the word-topic analysis of this project's validation test. This is because the number of discernible document topics is extremely limited compared to the number of possible word-word pairs of interest. Many single word-word pairs cannot be replicated by a corresponding word-topic pair, yet may be highly semantically revealing. It is in the ability to analyze these word-word pairs that word vector methods demonstrate their unique utility for political scientists over other existing text-as-data methods.

To demonstrate both the additive flexibility of the model and the semantic utility of word-word pairs over word-topic pairs, I used the trained chronological model outlined above to analyze a new word-word pair: "social" and "equality." Social equality is a complex, often euphemistic term of art used by American leaders after the Civil War; the words are often but not always used next to each other. Booker T. Washington, for instance, in his famous "Atlanta Compromise Speech" on race relations in the South, described how "in all things that are purely social we can be as separate as the fingers" while also explicitly referencing social equality in his assertion that "the agitation of questions of social equality is the extremest folly" (Washington 1895/1974). Because it is often used indirectly, vaguely, or obliquely, is a common word in other contexts, and when conceptually paired with equality references a wide range of substantive topics, tracking the relationship of social to equality over time is not tractable with either a topic model or word count approach.

Word vectorization, however, is admirably suited to the task of tracking the prominence of this "social" meaning of "equality". The proximity of "social" and "equality" in vector space directly reflects their semantic proximity in a given time slice in a given corpus. Figure 4 plots the cosine similarities of this pair over time.

The close proximity of social to equality – as high as gender and race in the first two eras – trails off as we approach the Civil Rights era. Scholars know social equality was a euphemistic stand in for fears of miscegenation and aversion to black-white intimacies of all types. Given this, the declining proximity of social to equality in the lead up to the Civil Rights movement and legal victories like *Loving v. Virginia* (1967), which struck down state laws banning interracial marriage, tracks expectations. That the decline of social equality began before the Civil Rights era might spark considerations about the importance of diminishing such social stigma as a *precursor* to changes in laws and institutions.

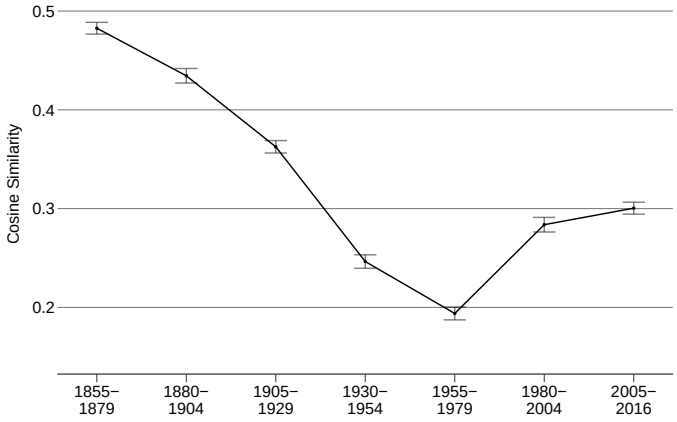

Figure 4: "Equality" - "Social" Cosine Similarity (with 95% confidence interval)

Figure 4 also revealed an unexpected feature of the corpus – the re-emergence of "social" as an important facet of equality's meaning in the most recent eras. Here the results sparked additional close reading of the corpus to determine what this re-emergence might signify. Is it a return to certain kinds of euphemistic racial stigmatization? Or is this a new valence of social equality that has emerged?

Close reading of the corpus in later eras suggests that the conjoint meaning of social and equality has expanded from a racial euphemism to also capture economic relationships. Phrases like "economic equality," "social issues," and "social justice" are used together, and social dimensions of equality are mentioned in articles about access to elite institutions like private golf clubs and Ivy League universities. At the same time, "social" has taken on a new negative valence, with some articles decrying an emphasis on social justice or social equality, where the "social" modifier – like in the Reconstruction era – serves as a euphemistic stand in for some vague, undeclared racial and economic agenda to which the writer is opposed.

8 Conclusion

By close reading, a political theorist can produce an admirable semantic analysis of how the meaning of rights discourse shifts between John Locke's *Second Treatise of Government* (with a mere 28 mentions of rights) and *The Federalist Papers* (with a still-manageable 152 mentions). While such an analysis is interesting in itself, and well within the capabilities of a single reader, we can easily imagine interesting cases which would exceed those capabilities. Computational methods can expand the universe of texts we can consider, and such methods have, among many other successes, already facilitated broader inquires into the meaning and development of words

like rights (see de Bolla 2013). Here, I have shown how `word2vec` can reveal the changing meaning of a single word like "equality" in a modest corpus of newspaper articles. In addition to tracking general trends in the meaning of equality as it relates to gender, race, and international relations, my analysis also highlights the granularity that is possible with word vector models: with my chronologically trained model, I show that the prominence of the idea of "social equality" inversely tracks racial progress, demonstrating unexpected commonalities between the post-Reconstruction era and the contemporary moment.

Word vectorization is a particularly promising computational method that has the ability to track the evolution of single words like rights or equality over time by tracking the cosine similarities of pairs of words. While computational linguists have shown how word vectorization of single corpora can produce impressive results on word analogy and synonym tasks, this project has highlighted the utility of word vector methods for more complex semantic tasks over time periods in which the cultural meanings of words evolve. By attention to details of implementation (bootstrapping, language stabilization, and chronological training), analysts can confidently discover semantic trends in corpora, a development which should be broadly interesting to scholars studying the evolution and history of ideas and concepts, and their relationship to politics, society, economics, and culture. Whether as an exploratory tool, as a means of validating close-reading insights from a small corpus on a much larger corpus of texts, or as a mechanism of producing free-standing analyses of vast corpora, word vectorization methods offer political scientists a useful strategy for treating texts as data to reveal changes in the meanings of concepts and ideas over time.

Funding

This work was supported by the Center for American Politics and Public Policy (CAPPP) at the University of Washington and by the National Science Foundation [#1243917].

Acknowledgements

I am grateful for the invaluable advice and feedback received at various stages of this project from Chris Adolph, Jeffrey Arnold, Andreu Casas, Ryan Eastridge, Aziz Khan, Brendan O'Connor, Brandon Stewart, Rebecca Thorpe, Nora Webb Williams, and John Wilkerson, as well as from participants at the Ninth Annual Conference on New Directions in Analyzing Text as Data (TADA 2018). Allyson McKinney and Molly Quinton contributed cheerful and diligent research assistance. This project was also improved by statistical and computational consulting provided by the Center for Statistics and the Social Sciences as well as the Center for Social Science Computation and

Research, both at the University of Washington.

References

Antoniak, Maria and David Mimno. 2018. "Evaluating the Stability of Embedding-based Word Similarities." *Transactions of the Association for Computational Linguistics* 6:107–119.

Arnold, Jeffrey B, Aaron Erlich, Danielle F Jung and James D Long. 2018. "Covering the Campaign: News, Elections, and the Information Environment in Emerging Democracies.".
URL: *osf.io/preprints/socarxiv/af9jq*

Bamler, Robert and Stephan Mandt. 2017. "Dynamic Word Embeddings." *Proceedings of the 34th International Conference on Machine Learning* pp. 380–389.

Blaydes, Lisa, Justin Grimmer and Alison McQueen. 2018. "Mirrors for Princes and Sultans: Advice on the Art of Governance in the Medieval Christian and Islamic Worlds." *Journal of Politics* 80:1150–1167.

Blei, David, Andrew Ng and Michael Jordan. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning and Research* 3:993–1022.

Bolukbasi, Tolga, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama and Adam Kalai. 2016. "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings." *CoRR* abs/1607.06520.

Box-Steffensmeier, Janet, John Freeman, Matthew Hitt and Jon Pevehouse. 2014. *Time Series Analysis for the Social Sciences.* New York: Cambridge University Press.

Bruni, Elia, Gemma Boleda, Marco Baroni and Nam-Khanh Tran. 2012. "Distributional Semantics in Technicolor." *Proceedings of the Annual Meeting of the Association for Computational Linguistics* pp. 136–145.

Caliskan, Aylin, Joanna J. Bryson and Arvind Narayanan. 2017. "Semantics derived automatically from language corpora contain human-like biases." *Science* 356:183–186.

de Bolla, Peter. 2013. *The Architecture of Concepts: the Historical Formation of Human Rights.* New York: Fordham University Press.

Firth, J. R. 1957. A Synopsis of Linguistic Theory, 1930-1955. In *Studies in Linguistic Analysis*, ed. J. R. Firth. Oxford, UK: Basil Blackwell.

Foner, Eric. 1998. *The Story of American Freedom.* New York: W. W. Norton.

Gallie, W. B. 2013. "Essentially Contested Concepts." *Proceedings of the Aristotelian Society, New Series* 56:167–198.

Garg, Nikhil, Londa Schiebinger, Dan Jurafsky and James Zou. 2018. "Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes." *Proceedings of the National Academy of Sciences* 115(16):E3635–E3644.
URL: *https://www.pnas.org/content/115/16/E3635*

Goldberg, Yoav and Omer Levy. 2014. "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method." *CoRR* abs/1402.3722.

Goldman, Merle and Elizabeth Perry. 2002. *Changing Meanings of Citizenship in Modern China*. Cambridge, MA: Harvard University Press.

Grimmer, Justin and Brandon Stewart. 2013. "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." *Political Analysis* 21(3):267–297.

Gurciullo, Stefano and Slava Mikhaylov. 2017. "Detecting Policy Preferences and Dynamics in the UN General Debate with Neural Word Embeddings." *IEEE Proceedings of the 2017 International Conference on the Frontiers and Advances in Data Science* pp. 74–79.

Hamilton, William, Jure Leskovec and Dan Jurafsky. 2016*a*. "Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change." *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* p. 2116–2121.

Hamilton, William, Jure Leskovec and Dan Jurafsky. 2016*b*. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change." *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* pp. 1489–1501.

Han, Rujun, Michael Gill, Arthur Spirling and Kyunghyun Cho. 2018. "Conditional Word Embedding and Hypothesis Testing via Bayes-by-Backprop." *Conference on Empirical Methods in Natural Language Processing* .

Harris, Zellig. 1954. "Distributional Structure." *Word* 10:146–162.

Heuser, Ryan. 2016. *Gensim word2vec Procrustes Alignment*. Github Repository at https://gist.github.com/quadrismegistus/09a93e219a6ffc4f216fb85235535faf.

Hopkins, Daniel and Gary King. 2010. "Extracting Systematic Social Science Meaning from Text." *American Journal of Political Science* 54(1):229–247.

Howard, Jeremy and Sebastian Ruder. 2018. "Universal Language Model Fine-tuning for Text Classification." *arXiv e-prints* p. arXiv:1801.06146.

Huang, Eric H., Richard Socher, Christopher D. Manning and Andrew Y. Ng. 2012. "Improving Word Representations via Global Context and Multiple Word Prototypes." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* pp. 873–882.

Iyyer, Mohit, Peter Enns, Jordan Boyd-Graber and Philip Resnik. 2014. "Political Ideology Detection Using Recursive Neural Networks." *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* p. 1113–1122.

Jockers, Matthew L. 2013. *Macroanalysis: Digital Methods and Literary History.* Champaign: University of Illinois Press.

Jurafsky, Dan and James Martin. 2009. *Speech and natural language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Upper Saddle River, NJ: Prentice Hall.

Kim, Yoon. 2014. "Convolutional Neural Networks for Sentence Classification." *arXiv e-prints* p. arXiv:1408.5882.

Kim, Yoon, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde and Slav Petrov. 2014. "Temporal Analysis of Language through Neural Language Models." *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science* pp. 61–65.

Kulkarni, Vivek, Rami Al-Rfou, Bryan Perozzi and Steven Skiena. 2015. "Statistically Significant Detection of Linguistic Change." *Proceedings of the 24th International Conference on World Wide Web* pp. 625–635.

Larson, Jeff, Julia Angwin and Terry Parris. 2016. *Breaking the Black Box: How Machines Learn to be Racist.* https://www.propublica.org/article/breaking-the-black-box-how-machines-learn-to-be-racist?word=Clinton.

Levy, Omer, Yoav Goldberg and Ido Dagan. 2015. "Improving Distributional Similarity with Lessons Learned from Word Embeddings." *Transactions of the Association for Computational Linguistics* 3:211–225.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* pp. 3111–3119.

Mikolov, Tomas, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *arXiv e-prints* p. arXiv:1301.3781.

Mikolov, Tomas, Wen-Tau Yih and Geoffrey Zweig. 2013. "Linguistic Regularities in Continuous Space Word Representations." *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* pp. 746–751.

Mimno, David. 2012. "Computational Historiography: Data Mining in a Century of Classics Journals." *ACM Journal on Computing and Cultural Heritage* 5.

Mosteller, Frederick and David L. Wallace. 1964/2008. *Inference and Disputed Authorship: The Federalist.* Chicago, IL: University of Chicago Press.

Nay, John. 2017. "Predicting and understanding law-making with word vectors and an ensemble model." *Plos One* 12.

Newman, David J. and Sharon Block. 2006. "Probabilistic Topic Decomposition of an Eighteenth-Century American Newspaper." *Journal of the American Society for Information Science and Technology* 57(6):753–767.

Pan, S. J. and Q. Yang. 2010. "A Survey on Transfer Learning." *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Pennington, Jeffrey, Richard Socher and Christopher D. Manning. 2014. "GloVe: Global Vectors for Word Representation." *EMNLP* 14:1532–1543.

Quinn, K., B. Monroe, M. Colaresi, M. Crespin and D. Radev. 2010. "How to Analyze Political Attention with Minimal Assumptions and Costs." *American Journal of Political Science* 54(1):209–228.

Rehurek, Radim and Petr Sojka. 2010. "Software Framework for Topic Modeling with Large Corpora." *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* May:45–50.

Reynolds, Noel B. and Arlene Saxonhouse. 1995. *Hobbes and the Horae Subsecivae.* Chicago, IL: University of Chicago Press.

Rhody, Lisa. 2012. "Topic Modeling and Figurative Language." *Journal of Digital Humanities* 2.

Rodman, Emma. 2019. "Replication data for: A Timely Intervention: Tracking the Changing Meanings of Political Concepts with Word Vectors." *Harvard Dataverse* V1. https://doi.org/10.7910/DVN/CGNX3M.

Rong, Xin. 2014. "word2vec Parameter Learning Explained." *arXiv e-prints* p. arXiv:1411.2738.

Rudkowsky, Elena, Martin Haselmayer, Matthias Wastian, Marcelo Jenny, Stefan Emrich and Michael Sedlmair. 2018. "More than Bags of Words: Sentiment Analysis with Word Embeddings." *Communication Methods and Measures* 12:140–157.

Saldana, J. 2009. *The Coding Manual for Qualitative Researchers.* Thousand Oaks, CA: Sage.

Schnabel, Tobias, Igor Labutov, David Mimno and Thorsten Joachims. 2015. "Evaluation methods for unsupervised word embeddings." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* pp. 298–307.

Turney, Peter and Patrick Pantel. 2010. "From Frequency to Meaning: Vector Space Models of Semantics." *Journal of Artificial Intelligence Research* 37:141–188.

Washington, Booker T. 1895/1974. Atlanta Compromise Speech. In *The Booker T. Washington Papers*, ed. Louis R. Harlan. Urbana: University of Illinois Press pp. 583–587.

Wilkerson, John and Andreu Casas. 2017. "Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges." *Annual Review of Political Science* 20(1):529–544.

Yang, Tze-I, Andrew J. Torget and Rada Mihalcea. 2011. "Topic Modeling on Historical Newspapers." *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities* pp. 96–104.

Yao, Zijun, Yifan Sun, Weicong Ding, Nikhil Rao and Hui Xiong. 2018. "Dynamic Word Embeddings for Evolving Semantic Discovery." *WSDM 2018: The Eleventh ACM International Conference on Web Search and Data Mining* .

You, Quanzeng, Jiebo Luo, Hailin Jin and Jianchao Yang. 2015. "Robust Image Sentiment Analysis Using Progressively Trained and Domain Transferred Deep Networks." *arXiv e-prints* p. arXiv:1509.06041.

Zhang, Ye and Byron Wallace. 2015. "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification." *arXiv e-prints* p. arXiv:1510.03820.