Unsupervised Document Clustering by Authorial Style through Network-Based Semantic and Syntactic Features

Kiko Ilagan
Stanford University
B.S. Biology
ilaganf@stanford.edu

Anoop Manjunath
Stanford University
B.S. Economics
amanjuna@stanford.edu

Vik Pattabi
Stanford University
B.S. Computer Science
vpattabi@stanford.edu

1. Overview

Natural language processing still heavily relies on vector space word representations as a key to understanding meaning and differentiating texts. While these representations remain important, especially as they are well suited for machine learning problems, recent work has looked to other possible representations of text, notably language as a network. Through identifying meaningful schemes to construct natural language as graphs, we hope to generate higher-level linguistic analysis focusing on more than just lexical meaning. Understanding how and when words or sentences interact and especially how these interactions change over time can generate key insights into often arcane questions such as "what makes a 'good' work *good*?"

2. Introduction

Much of NLP work has focused on techniques for text summarization, sentiment analysis, and textual similarity identification. Nevertheless, NLP techniques have incredible potential to answer fundamental questions about how people interact with language, and therefore, each other. For example – how to characterize different writing styles, especially across eras, subjects, or personal bias.

Although traditional NLP tools have relied on word embeddings to generate depictions of meaning, newer research has explored the potential for graphically representing text. Graph representations permit richer and more structured comparison of textual works, and might help supplement traditional semantic features with elements of syntactic information. This graph construction problem can be challenging: there are endless possible methods of representing text in a graph and it is critical to pick an algorithm that results in a meaningful graphical representation. Potential examples include connecting words with directed edges if they occur in sequence or connecting similar lexical substructures by similarity (for example, sentences). We hope to demonstrate the potential for combining network-based analysis schemes with traditional word embeddings to produce more robust and differentiated representations of texts.

3. Related Work

We discuss three papers that leverage graph algorithms to generate insight into natural language problems. Interestingly, all three papers propose applying network constructions to text summarization. Consequently, our intuition is that these graph construction methods might generate networks which better represent semantic content than syntactic content.

3.1. *LexRank: Graph-based Lexical Centrality as Salience in Text Summarization* (Erkan et. al) [4]

Erkan et. al address the challenge of text summarization, a classic natural language processing problem. Similarity metrics are taken between all sentences with sentences represented as one-hot vectors with dimensionality equal to the vocabulary size. We then treat sentences as nodes and construct edges between sentences based on similarity, with an edge existing if the similarity result is $\geq k$, a threshold hyperparameter. The edges are undirected, as similarity is a symmetric relation.

Two variants on PageRank are applied. First, the

authors construct a stationary distribution which represents the "importance" of each node. They call this base version "LexRank", although they further present an alternative called "continuous LexRank" which incorporates the previously discarded edge weights (similarity scores).

As the authors note, a poor choice of k could lead to a graph that is too dense or too sparse. This is a concern for us in our "bag-of-words" construction method, which we discuss further below. Furthermore, this graph construction scheme intuitively seems to focus more on semantic meaning than syntactic character; after all, the constructed graphs are ultimately relationships between similar "sets of meaning", and we might simply imagine a dense graph to indicate that the author repeatedly used different structures with similar meanings.

3.2. *TextRank: Bringing Order into Texts* (Mihalcea and Tarau) [6]

TextRank applies the "random surfer model" and scoring system from PageRank to graphical representations of text. For smaller lexical structures like words, the authors use co-occurrence to build the graph. The group experimented with the types of nodes included – creating graph of only certain syntactic elements (e.g. adjectives, nouns, etc.) or bipartite graphs of nouns to verbs. For larger structures like sentences, the group uses the system of "similarity" between sentences as applied by Erkan et. all's LexRank [4] to generate the graph.

We noted considerable opportunities for modification to better suit this algorithm to our task. Firstly, building a graph structure on the basis of co-occurance is naive. Related words may not be co-located (may be noun and object) and hence the hyperparameter of window size has an unduly large impact on the model performance. Considering the construction of the graph on the basis of sentence similarity, we further see that this approach is somewhat limited to sentence applications.

3.3. *An Approach to Graph-based Analysis of Textual Documents* (Bronselaer et. al) [2]

Bronselaer et. al also address multi-document summarization (MDS), although the focus of the paper rests primarily on considering schemes to construct networks from text in general. First, a piece of text is tokenized and a part-of-speech tagger is run. Then, the tokenized text is filtered by a "reclassifier", which eliminates words that don't strongly contribute to the information content of a sentence (determiners and adverbs per their heuristic). A graph is constructed such that "relationship" parts of speech (verbs, prepositions, and conjunctions) are edges and other words are nodes. Every node-edge-node in the text is added to the graph.

Significant semantic information is lost because connective words (e.g. verbs) are not represented as nodes in the graph. Despite the intuition between using them to connect objects in the graph, these words are also important to the overall meaning of the sentence (or document). In implementing this algorithm ourselves, we considered a variety of ways to incorporate this information.

4. Data

We are using two main datasets for our textual analysis, one of political speeches and another of politically-based sentences. We hope that by considering two document classes with significant size differences, we will be able to draw conclusions about the robustness of our approach across different snippets of natural language. For all of our analyses, we use 300 dimensional global word vectors (GloVe vectors [7]) trained on Wikipedia article text with a vocabulary size of 40000 unique tokens. Preprocessing for all datasets involves tokenizing the word files using the python package `nltk`. For each word, we check if there exists a valid embedding in the 40000 x 300 embedding matrix, and if not, we record the word as an UNK token.

Our first dataset is an archive of speeches delivered by presidents from Washington through Obama. The speeches are taken in plaintext form from [3]. The dataset consists of roughly 3.5 million words split between 962 speeches. Each president has roughly the same number of unknown words present (average of 0.002% of tokens were UNKS for each president). Given that records are better for newer presidents and older presidents have on average shorter speeches, we only considered speeches with less than 400 unique tokens, which yielded 312 speeches roughly evenly distributed across all presidents (when the number of speeches is normalized by speech length). This had the

secondary benefit of greatly increasing our processing time; running node2vec on the larger speech graphs was often intractable.

Our second dataset consists of sentences from the Ideological Book Corpus (IBC) [8]. The corpus contains 4,062 sentences selected from US congressional floor debates in the year 2005 annotated by political ideology. Of these 4,062 sentences, we considered 2,025 liberal sentences and 1701 conservative sentences, dropping the 600 neutral sentences. The mean number of unique tokens for each sentence in the dataset is 34.74.

Finally, one of our graph generation algorithms relies on the presence of part-of-speech tags on the text to construct relationships between nodes. For this task, we leverage NLTK's part-of-speech tagging functionality [1] which implements an off-the-shelf tagger using tags from Penn's Treebank tag-set [5].

5. Graph Models

We implemented 4 approaches to graphically represent our data, each aiming to capture different dimensions of the text meaning.

5.1. Text Bag Algorithm

This algorithm treats the input text as a bag of words with each unique word as a node. We calculate the similarity matrix S where $S[i, j]$ is the cosine similarity of nodes i and j using word2vec embeddings. For each pair of nodes (u, v), we draw an edge if the cosine similarity of their embeddings is in the 75th percentile of similarities in the document. Although this parameter was initially chosen arbitrarily, we found that minor variations from it did not substantially change the sparseness of the baseline graph (and thus the results of this baseline model). The initial value was selected given the example set from [4].

5.2. Sliding Window Algorithm

This graph generation scheme aims to better capture the sentence-level sequential relationships between words. We construct a graph with n nodes where n is the number of discrete tokens. We then iterate through the tokenized document, sliding a window of size 2 across the tokens; at each window step, the first and last element of the window are connected to each other. When the window encounters the end of a sentence, no connection is formed, meaning words are only connected by the window if they are in the same sentence. The intuition behind this technique is to link co-occurring word based on how we might read the text (from left to right); furthermore, sentences which share words will intersect through the shared word nodes, suggesting that more common words might become more central in this graph construction scheme.

Algorithm 1 Text Bag Algorithm

$V \leftarrow$ each unique word in document
$G \leftarrow (V, \emptyset)$
$E \leftarrow$ word embeddings
for $i, j \in V$ **do**
    $S[i, j] \leftarrow \frac{E[i]^\top E[j]}{||E[i]||*||E[j]||}$

for $i, j \in V$ **do**
    if $S[i, j] \geq 75\%$ of all similarity scores **then**
        $G.AddEdge(i, j)$

5.3. Part-of-speech Algorithm

The baseline algorithm uses word similarity, but it completely ignores other relevant features of a word, such as part-of-speech. We directly implement the algorithm from [2] as a competitor to the baseline. Importantly, [2] builds a directed graph incorporating the temporal nature of the sentences. However, our node2vec implementation only handles undirected graphs which prompted us to ignore this temporal feature during construction.

5.4. Sentence Chain Algorithm

The above approach uses parts of speech, but discards information about the meanings of the words that are being turned into edges. Additionally, it fails to maintain the higher-level chronological organization of a given work. Furthermore, although the window algorithm captures some element of word chronology, it oversimplifies this feature by ignoring the ordering of sentences, paragraphs, and other higher-level structures. To remedy these failings, the sentence chain algorithm first splits the work into its constituent sentences. It then connects the words within a given sentence both sequentially and using the same part-of-speech information as the above approach. We also create a meta-node for each sentence that connects to

Algorithm 2 Sentence Chain Algorithm

$T \leftarrow$ POS tagged document
$G \leftarrow (\emptyset, \emptyset)$
$E \leftarrow$ word embeddings
for each sentence i in T **do**
    $W \leftarrow$ all unique non-determiner words in i
    $G.addNodes(W)$
    connect words in sentence sequentially
    connect words that are separated by verbs
    $G.addNode(meta_i)$
    for word in sentence i **do**
        $G.addEdge(meta_i, \text{word})$

$S \leftarrow$ (num_sentences x embedding size) matrix
for each sentence i in T **do**
    $S[i] \leftarrow mean(E[neighbors(meta_i)])$
    $G.addEdge(meta_i, meta_{i+1})$
$sim \leftarrow SS^\top$
for each pair (i, j) of metanodes **do**
    if $sim[i, j] \geq 75\%$ of all similarity scores **then**
        $G.AddEdge(i, j)$

each of the words in the sentence, and we connect these meta-nodes sequentially according to the sentence order of appearance in the work. As a final step, we then then approximate a "meaning" for each sentence by averaging the word embeddings of the words in the sentence. We reasoned that the mean would be more robust to sentence lengths, since length could be captured by the degree of the sentence's meta-node. We connect the meta-nodes of sentences that have a similarity (measured by dot product) in the 75th percentile or above of sentence similarities within the document.

6. Analysis Techniques

We present two elementary analysis techniques to extract meaning from the constructed graphs. We also include a third scheme which simply concatenates the vectors from the following two schemes.

6.1. Meta Node Embedding

Once the graph is generated, we insert a supernode into the new graph that is connected to every other node. We then take the node2vec vector of that node to represent a style vector for the overall graph. The node2vec parameters were determined after a short empirical search and involve 10 random walks with $p = 1$, $q = 3$, of length 80. The output dimension is 128. In constructing this feature vector, we also tested the addition of both average clustering coefficient and average degree (sampled from 100 randomly selected nodes) as metrics in our style vector. However, these measures, having little variance across the data, were dropped from consideration as part of the style vector.

Intuitively, we want our calculated style vector to somehow extract relevant style information from the constructed graph. A "supernode" connected to part of speech components might do this, as a node2vec representation of this supernode will incorporate information about the directional relationship (or lack thereof) between different textual objects. Given our aim to capture a vector representation of the general graph structure, we chose our node2vec parameters to encourage the random walker to explore further away from the supernode and deeper into the true graph.

6.2. Node Centrality Featurization

Another graph featurization we developed used eigenvector centrality to compute the top 5 most central nodes for each document graph. We then averaged the embeddings of these central words, resulting in a 300-dimensional feature vector. Among all the possible centrality measures (harmonic, between-ness, etc.), we chose eigenvector centrality to better emulate the output of PageRank style random walks on our generated document graph. Our intuition was that these random walks might parallel how an individual would read a document, especially on the non-Text Bag models which incorporate word order in graph generation.

We were initially concerned that centrality might be less meaningful simply because of inherent language variation over time (making any set of 5 words reasonable features). For example, if presidents in 1800 used a radically different vocabulary set from modern ones, the least central nodes in a graph might be just as telling. However, our intuition about the contribution of centrality was justified when testing against a null model (described more in subsequent sections).

Importantly, we ran a modified version of the node centrality scheme on the Sentence Cluster graphs.

Given that these graphs included additional 'sentence nodes' which were connected, we selected the top 5 word nodes by centrality after filtering out all non-word nodes in the centrality rankings. Furthermore, across all graph types, the node centrality featurization was calculated before the meta node featurization (to avoid the centrality effects of the meta node).

7. Experimental Methodology

We took several steps to analyze the generated "style vectors" in light of the underlying cluster distribution in the datasets. For the key analysis, we clustered variants of the feature vectors above and compared these results to our underlying ground truth. Specifically, we ran a K-means clustering algorithm (the `sklearn` implementation) on an array of document features while specifying the underlying number of clusters; this was determined from our knowledge about the datasets.

We ran this K-means clustering approach for each dataset across 4 different feature representations: just meta node featurization, just node centrality featurization, random node embedding featurization, and a feature vector concatenating meta node and centrality features. This selection was designed to confirm or refute our hypothesis that some combination of structural and meaning-based features would best capture cluster style (with meta node representing syntactic structure and centrality representing meaning). In the random selection scheme, we randomly selected 5 nodes from the graph to construct a meaning embedding, as opposed to selecting the 5 most central nodes; this served as a null model against which we could validate the contribution from the centrality features.

For both the IBC and presidents datasets, we chose to search for $k = 2$ clusters in our text data. This was a clear choice for IBC, as we hoped to expose differences in left-leaning vs. right-leaning sentences. Of the possible cuts of data in the presidential speeches, we initially considered three options: president, political party, and time of presidency. With respect to the former, we felt there might not be a strong inherent clustering – after all, many presidents likely don't have profoundly different topical focuses and syntax across their full repertoire of speeches (e.g. George H.W. Bush and Ronald Reagan might be similar, or Jefferson and Madison). We felt political party might also be less promising for several reasons. The history of political parties in America is complicated - some parties no longer exist (e.g. the Whigs), and a strange phenomonena post-labeled the party switch happened during the end of the 19th century and beginning of the 20th century where the major parties came to adopt each others' values. Furthermore, we suspected syntax changes might be less evident across party lines; there's no reason to suppose Democrats holistically use shorter sentences or more nouns for example. We felt a party clustering scheme might force us to place more weight on speech meaning in direct contradiction to our original curiosity regarding the addition of stylistic or syntactic structure.

On the other hand, we felt time clustering was well suited to leveraging the combination of syntax and semantics; after all, we might imagine speech meaning to change greatly locally despite the fact that syntax changes gradually. Nonetheless, these gradual syntax changes aid to differentiate speeches on common topics (e.g. the economy) which might occur in any time period. We opted to try and identify two speech clusterings – before and after the year 1900. This was not an arbitrary choice, as the median year in our dataset (labeling each president by the year they took office) was 1898. 1900 seemed a reasonable choice in this context given that it was also an election year. Furthermore, in hindsight, this specific clustering problem is especially interesting given the events of the early 20th century during which America became a more influential power abroad (likely reflected in the dataset). Potential future work (fleshed out in a subsequent section) might investigate more granular clusterings (perhaps via historic era).

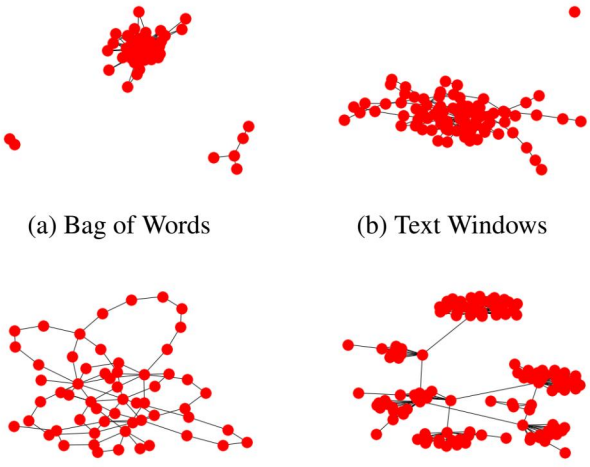
We also learn a t-distributed Stochastic Neighbor Embedding (t-SNE) for each speech style vector in two dimensions. Before the t-SNE, we perform PCA dimensionality reduction to 10 principal components. This initial dimensionality reduction is recommended as part of the pre-processing before t-SNE [9]. Although this does not leave a quantitative measure, the t-SNE visualizations captured the clustering we were looking for and helped us fine-tune our model parameters as we worked toward a final model.

Finally, we note that we filtered out graphs with $|N| \geq 400$ during the main phase of experimentation, leaving us with in total 312 presidents graph (having

eliminated 650 graphs). However, we present a small experiment utilizing the node centrality featurization on the full dataset (all graph sizes) as well.

8. Results

We used all 4 described graph generation algorithms to construct graphs for every speech in our corpus. The structures for one particular speech, President Franklin Delano Roosevelt's "Declaration of War on Germany", delivered on December 11, 1941, are presented below in Figure 1 using a basic force-directed layout for visualization:



(a) Bag of Words (b) Text Windows

(c) Text Parts of Speech (d) Sentence Chains

Figure 1: Graphical representations of FDR's declaration of war on Germany using different graph construction algorithms

We can see that each algorithm generates a visually distinct structure for the same speech. In particular, the bag-of-words and text windows algorithms appear to result in tightly clustered components, whereas the parts of speech and sentence chain algorithms have a more spread out patterns of connection and clustering, as expected.

From general observation, we see that the Textbag tends to create a graph with several (on average 7-8) strongly connected components. There is one central strongly connected component surrounded by satellites which typically consist of 10-20 nodes. The text windows graphs appear to generally be strongly connected; in rare cases, one or two nodes orbit the central SCC. Our intuition is that these nodes represent short sentences which minimally intersect the main content of the speech - perhaps exclamations or strong interjections. The graphs generated using parts of speech universally consist of a single strongly connected component. Finally, as expected, the sentence chain graphs are all single strongly connected components; this is unsurprising as the algorithm explicitly connects each sentence meta-node together in sequence; even a minimal number of extra similarity connections will link any two words through the sentence node chain.

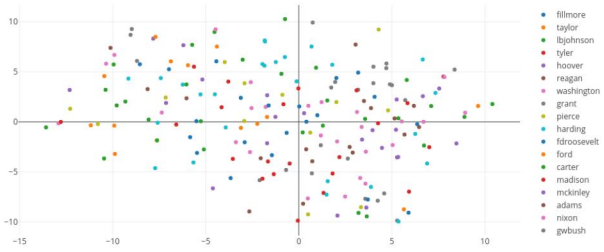


Figure 2: t-SNE plotting of supernodes derived from a graphical representation of each presidential speech (generated using the baseline Text Bag algorithm). Each individual colored point is a speech

As we can see from Figure 2, the baseline graph generation algorithm does not show any organization, clustering or otherwise, when the meta-node analysis is used. This result is not entirely unsurprising – by connecting nodes with high cosine similarity within a speech, we are only capturing information about how often a given author uses related terms within a single speech. We get no information about the actual content of the speech, nor do we necessarily capture anything about how the words are connected and related to each other. In short, this approach is not conducive to extracting a meaningful graphical representation of a written work, despite receiving endorsement from [4].

Table 1 displays our accuracy results on predicted clusterings of the Presidents dataset using three different node centrality measures. Each of the centrality measures (eigencentrality, between-ness centrality and harmonic centrality) were implemented from `networkx`. Our goal for testing all three features despite the theoretical fit of eigencentrality was to explore the viability of different centrality metrics on

different graph structures. Unfortunately, we see no clear winner, with each centrality measure performing the best on a different graph model. Nonetheless, between-ness centrality and eigencentrality typically performed the best. We were surprised at the sentence chain result indicating that between-ness centrality yielded the greatest improvement over the random node features and was holistically the best; intuitively, high-betweenness nodes on the sentence chain graph should be the discarded sentence meta-nodes.

We were further interested to see the overall performance with regards to clustering accuracy of the different graph models. Unsurprisingly, both sliding window and text bag yielded the worst results, which in effect were only slightly better than a random cluster assignment (intuitively the worst case assignment would mis-classify roughly half of the speeches, especially given that we selected the dividing line based on the median speech year). We expected that centrality might be less meaningful on these graphs, and especially the text window graph, as this graph scheme did not eliminate determiners or other frequently used generic words (e.g. 'to') that would typically be central given their high usage.

On the other hand, the part-of-speech and sentence chain model both performed better, yielding equivalent best scores of 0.708. Comparing the centrality results against the featurization from 5 randomly selected nodes, it was clear that centrality information was meaningful in the clustering. Furthermore, the better performance of part-of-speech and sentence chain even in the random node scheme indicated these graphs were more information-rich. We felt the sentence chain graph was at a disadvantage with respect to the centrality featurization, as much of its structure came from the meta-nodes which have no previously-determined embeddings. Consequently, we were not surprised to see lower eigen-centrality and harmonic centrality performances here, as these metrics are inherently biased toward selecting central nodes (meta-nodes) which we then discarded from the sorted centrality list.

Given the above results, we elected to continue using eigen-centrality as our centrality featurization metric. We felt it better represented how a human might read text, and we felt less confident about the performance of between-ness centrality in the sentence chain model overall. We applied a Borda scoring rule to the performance placements of each metric under each graph scheme, which further reinforced our choice of eigen-centrality in the face of these inconsistent results.

Having seized upon eigen-centrality as our centrality measure, we proceeded with our more complete analysis regarding the utility of combining centrality and meta-node node2vec to separate speech style. The results of our analysis is presented in Table 1.

As we can see, the node2vec representation of the graph meta-node appears to add little value to separating the presidential speeches by time. Its exact value varies with the graph representation – it has minimal impact for graphs generated using parts of speech and text bags, however it has a much more substantial impact on graphs generated by considering sliding windows over the text or sentence chains. Regardless, taking the mean of the word vectors of the 5 most central words (by eigenvalue centrality) in the the parts of speech representation of the speeches produced the cleanest separation of speeches by time. Adding the node2vec vector of the metanode adds a marginal 0.5% on the classification accuracy, making it the most effective analysis technique for each graph generation algorithm. These numerical results can be corroborated by visual inspection. In Figure 3 we present the t-SNE embedding of the vectors generated from each of the analysis techniques (eigen-centrality, meta-node node2vec, etc.) on the text graphs generated with parts of speech, with each speech being colored according to its presentation date.

We see that the vector representation of text graphs produced from eigenvalue centrality carry valuable information regarding the style (proxied by time period) of the speeches. The centrality vectors, without and in combination with the metanode node2vec vectors, have embeddings that order with regards to time. In particular we see a gradient with regards to time of speech along t-SNE axis 1; this occurs in both the plot with only eigenvalue centrality and the one using the concatenated vector of metanode node2vec and centrality.

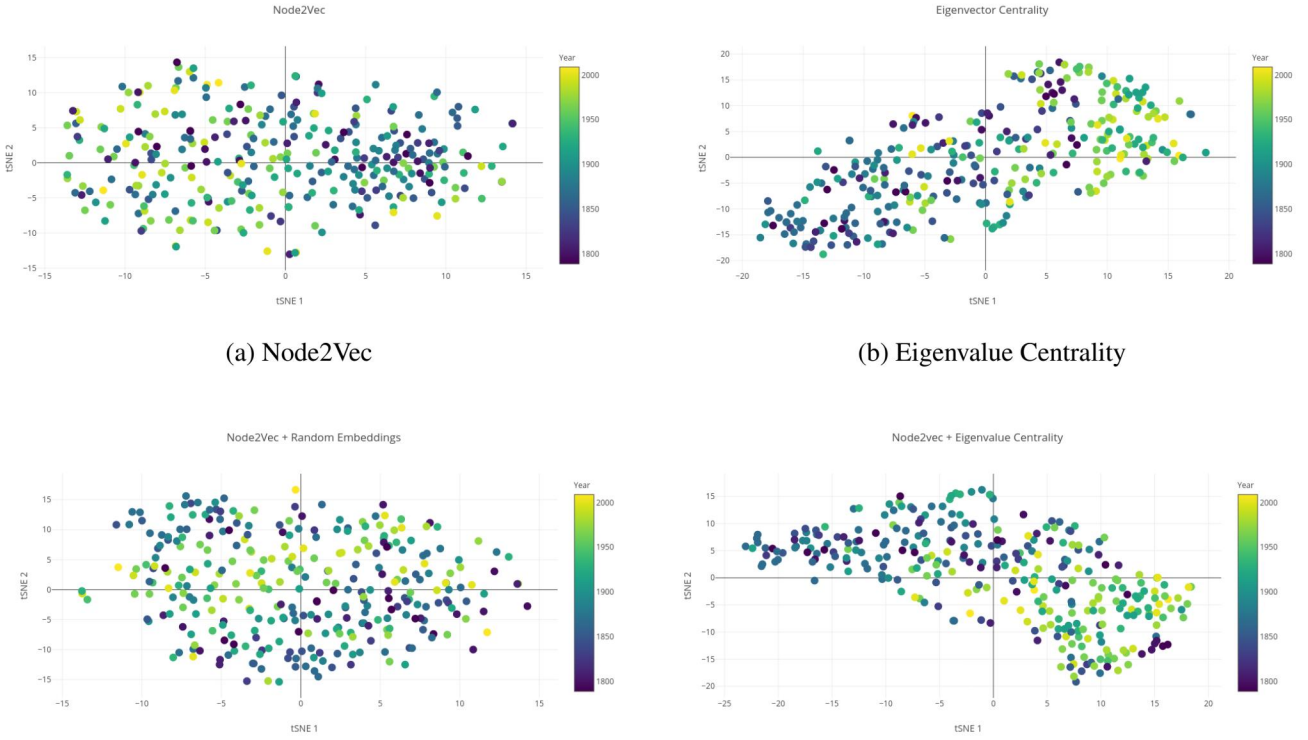
We also ran experiments on the IBC (Ideological Books Corpus) sentence dataset, testing all four graph construction methods with 4 featurization schemes. Table 3 displays these results, which, unsurprisingly,

	Part-of-speech			Text Bag			Sliding Window			Sentence Chain		
	Eigen	Between-ness	Harmonic	Eigen	Between-ness	Harmonic	Eigen	Between-ness	Harmonic	Eigen	Between-ness	Harmonic
Centrality	**0.708**	0.689	0.696	0.558	0.548	**0.587**	0.529	**0.567**	0.526	0.660	**0.708**	0.590
Random nodes	0.657	0.593	0.622	0.542	0.583	0.545	0.561	0.542	0.593	0.587	0.615	0.587

Table 1: Accuracy of the k-means clustering on nodes chosen through centrality measures vs. the null model, using 3 different centrality measures.

	Part-of-speech	Text Bag	Sliding Window	Sentence Chain
Eigen-centrality	0.71	0.52	0.53	0.66
Meta-node node2vec	0.52	0.55	0.56	0.58
Random node selection	0.64	0.51	0.51	0.57
Node2vec + centrality	0.71	0.58	0.58	0.70

Table 2: Accuracy of k-means clustering on different graph featurization schemes for the presidential speeches dataset.



(a) Node2Vec

(b) Eigenvalue Centrality

(c) Random Word Vectors

(d) Node2Vec and Eigenvalue Centrality

Figure 3: t-SNE plots of the style vectors derived from the 4 clustering methods. Each dot is a speech, and is colored by the start year of the president who delivered it

are relatively poor. IBC documents were each sentences, so the graphical representations were likely too small to extract significant meaning for the clustering. Interestingly, there was no clear feature scheme which yielded the best results; however, it is clear that the combining the node2vec and centrality features was less meaningful on the IBC texts, a result that contradicted the outcome from the presidential speeches. We suspect this may be because stylistic textual information is less dense at a sentence level, or less consistent

	Part-of-speech	Text Bag	Sliding Window	Sentence Chain
Eigen-centrality	0.57	0.52	0.5	0.54
Meta-node node2vec	0.51	0.52	0.54	0.59
Random node selection	0.54	0.54	0.51	0.5
Node2vec + centrality	0.57	0.51	0.5	0.54

Table 3: Accuracy of k-means clustering on different graph featurization schemes for the IBC dataset.

across different data points in a given cluster.

9. Discussion

The main challenge in this project was, naturally, finding a good way of formalizing human intuition for what constitutes style. There are many different potential approaches for connecting words in a document to turn it into a graph, but only some of these approaches are appropriate for our problem.

Our experiments showed that graph construction approaches that relied more on grammatical structure outperformed approaches that simply relied on word vector similarity. Additionally, the accuracy of our approaches increased with the size of the input speeches (IBC vs. presidential speeches), most likely because longer speeches were naturally able to exhibit a greater diversity of grammatical structure which led to a richer graphical representation.

In particular, we saw that node centrality measures worked particularly well with the part-of-speech graph generation algorithm. This result can likely be attributed to the emphasis that the algorithm puts on words on either side of connective strings – it makes sense that if we treat connective words (e.g. verbs and verb phrases) as edges, then the most central or important words will be the ones that are proximal to the most trafficked connectives.

On the other hand, meta-node embeddings were not as impactful as we had originally anticipated. The approach actually led to worse accuracy than the null model with the part-of-speech algorithm, and it provided only small improvements for the other models. The results do show a slight synergistic effect between meta-node embedding and centrality on the presidential speech dataset with all algorithms except for part-of-speech. Likely the embeddings had a larger impact on the non-grammatical graph generation algorithms (text bag and sliding window) simply because the graphs themselves were less reflective of the underlying structure, making centrality approaches less effective by comparison – note that the absolute improvement over the null still remains fairly small. It is also possible that the node centrality measures outperformed meta-node embeddings due to the mismatch in dimensionality – since the eigencentrality vector is 300 dimensional (based on the word embedding size) while the node2vec embedding is only 128, there is a potential mismatch in expressivity. This could have propagated through the K-means clustering implementation we used to yield better results for centrality. We might compare the performance of a PCA of centrality against node2vec in the future to examine this possibility.

Ultimately, our approach did manage to capture a shift in the rhetoric of the presidential speeches pre- and post- 1900. Interestingly, the most central/between words in the pre-1900 speeches were words such as “State” or “united” whereas many of the corresponding words in the post-1900 speeches had to do with overcoming adversaries. We speculate a few possibilities for this shift: perhaps pre-1900 speeches relied on appeals to central authority, but as institutional trust began to falter closer to the present day, speech makers found that unification against a common enemy was more compelling. Alternatively, it may be the case that America engaged in more belligerence post-1900: World Wars I and II, the Cold War and its resulting proxy wars, the Korean and Vietnam Wars, and the War on Terror are all examples that readily come to mind. It may have been the case that America’s legitimacy needed no internal validation once it became a major player on the global stage.

Investigating cases of misclassification yielded interesting insights. One commonly mis-classified speech was Zachary Taylor’s “Message Regarding Newly Acquired Territories” delivered in 1850. Although we might suspect this speech to greatly differ from more modern ones, sample sentences con-

tradict this intuition. For example, Taylor said "It is undoubtedly true that the property, lives, liberties, and religion of the people of New Mexico are better protected than they ever were before the treaty of cession." This rhetoric is not fundamentally stylistically different from that of a modern president; furthermore, it is not implausible to imagine some of these words (e.g. property, lives, liberties, protected) present in recent political dialogue. From inspecting these failure cases, we suspect the clustering scheme was unreliable when *both* syntactic and meaning based features overlapped across the time split. Perhaps the history of presidential rhetoric is not as diverse as we might expect; Americans today likely want similar guarantees from their government as those in previous eras.

While this observed divide in content is intriguing, whether or not it reflects a true shift in "style" remains in contention. Our sense was that the approaches we laid out captured important content information, but it seems doubtful that the extracted information was particularly stylistically idiosyncratic with respect to any of the individual speech writers. Linguistic style represented in the graphs was certainly valuable as the basis for identifying meaning through centrality, but the lack of strong results from the node2vec metanode suggests that our style graphs were not strongly distinct independent of node meaning.

10. Conclusion

We have presented several different graph generation and analysis techniques that aim to capture a meaningful representation of authorial style. As we expected, the approaches that incorporated both syntactic (using grammatical structure) and semantic (using word embeddings) information were strongly able to detect meaningful clustering in the input data. These techniques perform better on larger input speeches, and they are able to find important words that align with human intuition; the method was robust enough to identify reasonable clusters without supervision.

It was inherently difficult to measure success for this endeavor, as there does not seem to be much consensus on what even constitutes style. Our approaches did capture style in a broad sense – we were able to see that the particular appeals to authority or emotion made in the speeches we analyzed changed over time. However, this rough conception of style mostly serves as a vehicle to present meaning, as opposed to treating style as an equal facet of the full text.

To that end, we would be interested to see how these approaches might cluster works by a range of literary figures, who we suspect could produce more differentiated graph structures. Alternatively, this analysis could be pushed further through a greater focus on relationships between authors or themes across time period; investigation into this area could help uncover attribution or influence links or help define better features to strengthen the K-mean clusterings. In general, our original goal of pinning down a satisfying representation of a particular author's writing style through networks has eluded us, leaving much room for further study. The full source code for this project can be found at https://github.com/amanjuna/textnet.

References

[1] S. Bird and E. Loper. Nltk: The natural language toolkit. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[2] A. Bronselaer and G. Pasi. An approach to graph-based analysis of textual documents. In *8th European Society for Fuzzy Logic and Technology (EUSFLAT-2013)*, pages 634–641. Atlantis Press, 2013.

[3] B. D.W. Corpus of presidential speeches. `http://www.thegrammarlab.com`.

[4] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[5] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[6] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[7] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[8] Y. Sim, B. D. L. Acree, J. H. Gross, and N. A. Smith. Measuring ideological proportions in political speeches. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 91–101, 2013.

[9] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.