Extracting References from Political Speech Auto-Transcripts

Brandon Roberts
brandon@bxroberts.org
Austin Cut Research Group
Austin, TX 78704, USA

ABSTRACT

This paper presents an unsupervised method for counting references in noisy auto-transcribed political speeches. Transcriptions are vectorized using learned embeddings which are then clustered using k-means resulting in groups of words which represent highly granular, specific topics within the text. Words from each cluster are then extracted from each transcript, counted, and arranged for time-series analysis. The approach finds semantically coherent topics representing specific references despite transcription inaccuracies. We use this framework to extract references from over 400 political speech transcriptions from a 2016 U.S. presidential campaign.

CCS CONCEPTS

• **Computing methodologies → Information extraction**; **Machine learning**; **Unsupervised learning**; **Cluster analysis**;

KEYWORDS

Document Clustering, Word Embeddings, k-Means, Time-Series

1 INTRODUCTION

Machine-generated speech transcriptions are noisy documents which often contain metadata such as location and date. Political speeches are typically concerned with a relatively small set of themes which are frequently repeated. They are also filled with sporadic references to specific people, places, and events which are important in the moment. As political speech becomes distributed more commonly using digital means, particularly via online video, consuming this information in the form of machine-transcribed text becomes more commonplace. Discovering highly specific topics in these documents is a difficult, yet important task that is complicated by many factors.

Probabilistic topic models such as latent Dirichlet allocation (LDA)[3] and its variants[2, 17] represent documents as a collection of topics, each in varying proportions. But the topics generated by these models are often found to contain loosely or totally unrelated terms when manually inspected by those with domain knowledge. When topic modeling is applied to supervised tasks like classification, these probabilistic methods work well and have quantifiable results. In unsupervised applications, robust evaluation methods are still lacking and is an open area of research[5, 14, 24, 25].

Obtaining topics which are made up of references to specific events, without large numbers of unrelated terms, requires using high numbers of topics. Unfortunately, traditional topic models like LDA tend to generate increasingly incoherent topics as the number of topics increases[20]. Further, Mimno et al. (2011) note that topic size itself is a predictor of quality and that smaller topics are generally of low quality. To solve this problem, we propose using a series of simple and well understood techniques.

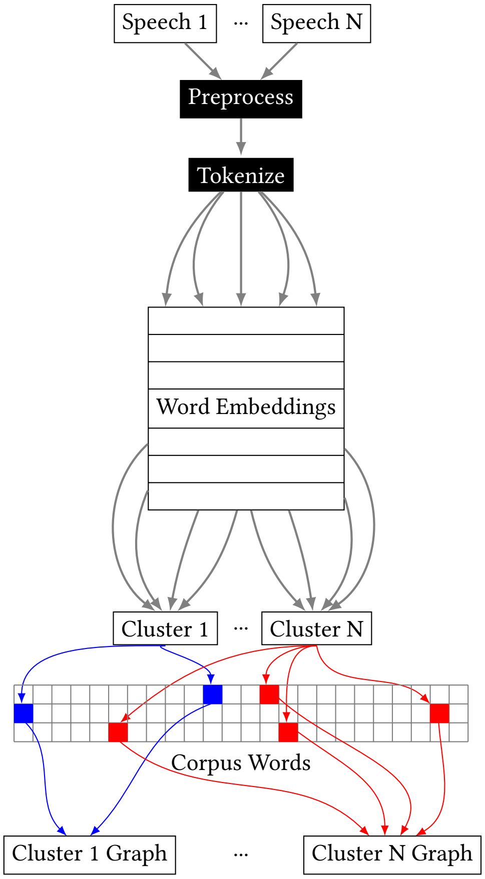

Figure 1: Model architecture for analyzing time-series themes from auto-transcripted political speeches.

Analysis of political text, particularly when using transcriptions of speech, is complicated by two main factors: 1) audio-to-text transcription errors and 2) the repetition of a small set of political themes spanning large majorities of the dataset intermixed with short lived event-related references. Our framework addresses both.

We present a framework that builds word embeddings from speech transcripts and uses spherical k-means clustering[8] to extract semantically coherent and highly granular topics from political speech transcripts (also referred to as *documents*)[1]. We can

[1] We refer to a single political speech transcript, in its entirety, as a *document* throughout this paper. Our dataset is made up of a collection of documents, each representing the machine-generated transcription from the audio of a speech.

DS+J, Aug 2017, Halifax, Nova Scotia, Canada
2017.

then take the word-cluster relationships, combine the associated temporal information about each document, and conduct a time-series analysis. This technique is fast, scalable (due to the efficiency of learning word vector representations), and manages to extract semantically cohesive topics from unstructured text. Further, the framework is robust to the specific number of topic clusters selected and incorporates the presence of numerous transcription errors.

In the data journalism field, much attention is spent working with manually transcribed and tabulated political speech[11]. Prevalent strategies for analyzing this type of data include using topic models and, predominantly, simple word counts. We propose and demonstrate a method to automate such work.

2 EXTRACTING UNIQUE TERMS FROM TRANSCRIPTS

2.1 Complications of Automated Speech Transcripts

Political speech, particularly in the context of a campaign trail, typically contains topical references which are repeated frequently over periods of time. New topics are introduced, often in response to an event or strategy, and others are phased out. Unlike other popular datasets for topic analysis, individual political speeches share many of the same words and themes, but references to politicized events, often appear suddenly in a short period of time.

Automatic speech transcriptions, in particular, present another set of challenges for text analysis. Such documents tend to be noisy, error-prone and contain both word misspellings and homonyms. These issues increase word count and dimensionality (particularly for bag-of-words models) without providing useful information. Strategies like stemming can alleviate some of this, as in the case of a simple typo or misspelling, but complications like homonyms cannot be remedied using this technique.

2.2 Term Extraction

In our framework, text preprocessing involves a series of simple transforms: lowercasing and removing non-printable characters, punctuation, URLs and any machine-readable tags. We then normalize whitespace, eliminate numeric-only tokens and transcription artifacts like audience cues (transcripts often embed reactions such as applause surrounded by brackets or asterisks). Words are tokenized by splitting documents on whitespace.

In addition to simple cleaning and tokenization, we have several other preprocessing requirements: 1) the need to strip out stopwords without relying on a hand coded list, 2) the need to be cautious and not eliminate misspellings or homographs, and 3) the desire to eliminate words unlikely to be topically important (such as prepositions). Since our overall framework allows us to quickly parse the resulting word clusters, the presence of potentially useless groupings is assumed to outweigh the cost of leaving out potentially valuable ones. We found that using Term Frequency Inverse Document Frequency (TF-IDF) scores was a simple way to achieve these goals.

We compute TF-IDF for all words in our dataset and then score, rank, and eliminate extremely common words with low TF-IDF scores. We found that retaining 75% of top-scoring words from each document was sufficient to limit the number of stopwords without removing potentially valuable ones. Using TF-IDF also eliminates the need for sentence boundaries, which are not present in audio transcripts, and are often required by sentence level part-of-speech extractors[7, 15].

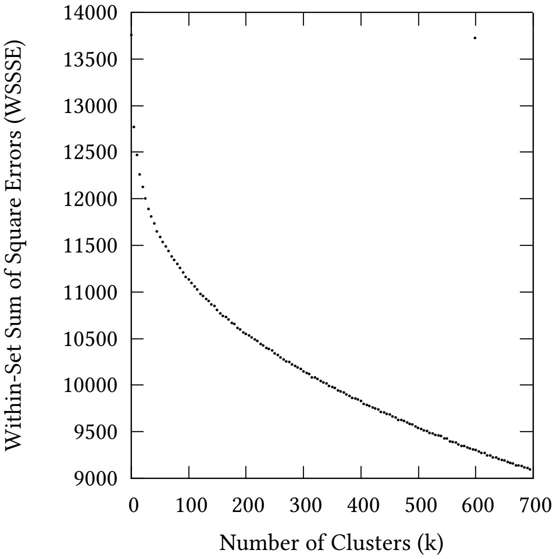

Figure 2: WSSSE vs. number of spherical *k*-means clusters.

3 WORD EMBEDDINGS

Word embeddings are vector representations of words[6, 18], learned by training a simple feed forward neural network on words and the context in which they're found within a set of documents. These vector representations have been shown to successfully capture the semantics of language and can be manipulated using algebraic operations[18]. The canonical example being when one subtracts the vector for the word *man* from the vector for *king* and adds the vector for *woman*, the result is a vector closest to the representation for the word *queen*.

An extension of this framework[4] learns character n-gram representations, which are then summed, resulting in word vectors. This proposed method is intended to capture not only contextual information, but also how words are individually formed. Since words are made up of characters, modeling word vectors as the sum of its contents allows us to better model rare words, uncover semantic similarities between homonyms, and even infer the representations of out-of-vocabulary words. Since speech transcripts are inexact mappings from audio to text, these properties are useful for modeling the semantics of speech transcriptions. For this reason we have chosen fastText[2], an implementation of this n-gram word embedding model[4], to vectorize our data. The model is trained on the entire corpus, not simply the TF-IDF extracted terms. Once complete, we take our extracted words and obtain a set of unique vectors for each political speech document.

[2] https://github.com/facebookresearch/fastText

4 CLUSTERING OF WORD EMBEDDINGS

Clustering is a fundamental task in all of data science, including unsupervised text analysis, and k-means is a well-known algorithm in this pursuit. Classic k-means uses Euclidean distance, which was found to be a poor metric for measuring the similarity of word embeddings. Considering this, we would like to use cosine similarity as a distance metric, as is typically done when comparing these types of representations[18, 19]. Therefore, we propose using spherical k-means[1, 8], which projects word vectors onto a high-dimensional, length normalized hypersphere and partitions them into clusters using cosine similarity.

We gather the unique word vectors and cluster them using spherical k-means. This results in semantically cohesive collections of words from our auto-transcribed political speeches. Further, we can take the centroids of each cluster, go back to the embeddings from the entire corpus, and find the words nearest to each cluster. This serves several purposes: 1) automatic ranking of words within our cluster, 2) quick summarization of the cluster and 3) allows for discovery of words which did not get extracted by TF-IDF, yet otherwise naturally belong in our word group. This final use of centroid-word-similarity was not explored rigorously, but did appear to be promising technique.

In our analysis of the cluster assignments of our political speech text, we found that our model was able to pull out very finely granular, highly semantically related topics from the 2016 U.S. presidential campaign. The number of clusters was determined by plotting the within-set sum of squared errors (WSSSE) over a range of k and then searching for an inflection point where it appeared that significant reductions of WSSSE slowed. Another technique explored was using pointwise mutual information to identify similar inflection features over a range of k. We cover this in Section 6.1. The range of possible k was determined through a simple heuristic of centering our search space approximately near $\sqrt{vocab_{unique}}$.

Figure 2 shows the plot of WSSSE for our spherical k-means model while $1 \leq k < 700$.

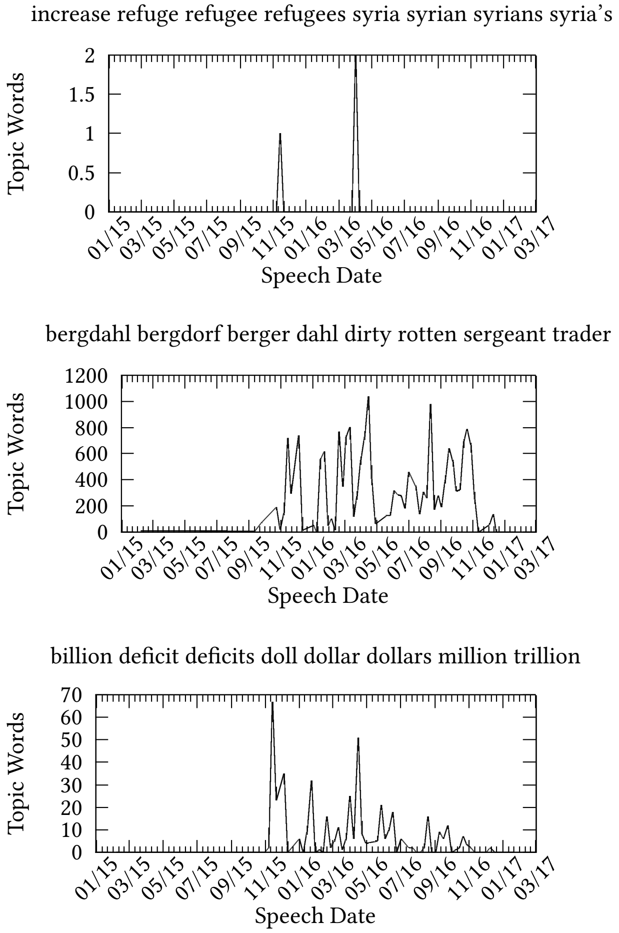

Figure 3: Example clusters from our dataset. Title for each plot consists of the cluster words. Time-series plot points are grouped per-week.

5 TIME-SERIES ANALYSIS

With topic clusters uncovered from our data, the date of each speech is then extracted from speech metadata. For each speech, we take each word from every cluster and count how often and when it appears across all speeches. Words within a cluster are then summed, producing the topic frequency per speech.

Cluster-topic frequencies are summed by week, which provides a smoothing effect. The data is then plotted as a time-series line chart and annotated with the centroid summary and the cluster members. Charts were inspected visually for clear trends and outliers, of which there were many. Section 6 contains our findings.

The frequentist approach was chosen because it embeds the intuition that the more a politician references something in a timespan, the more important it is. We experimented with normalization by document length, and represented topic mentions as a percentage of total words available, but found this approach resulted in extremely noisy charts that were easily dominated by short transcripts.

6 RESULTS

Our technique was performed against a real-world dataset of 413 auto-transcripted speeches from a 2016 U.S. presidential campaign. The data spanned approximately 177 hours of political speech from a single candidate between February 2015 and January 2017. Speech transcriptions were extracted from YouTube, which uses a deep neural model to perform the audio transcription into English, among many languages[16].

Transcripts are available in the Web Video Text Tracks format (WebVTT) as caption files which were converted to a plain text format. Preprocessing, as described in Section 2.2, was performed. The top 75% of TF-IDF ranked words were obtained from each transcript, resulting in 15,558 unique terms[3]. A fastText skipgram model was trained on the entire corpus (consisting of 1,031,901 total words) with word vector dimension, char n-gram min, and

[3] Note that this is the number of unique words remaining after TF-IDF stopword removal.

MSE	Cluster Terms
1.928	advertising brewing campaigning drawing fundraising knowing pleading screwing sewing showing speaking spear specially speech speed speeding spelling spells spend spends spent spewing swing swinging wing wingin winning
1.974	social socialism socialist socialize socialized socially socials socio
2.333	crew crews crude cruise cruises cruz cruz's favor kasich liar lies lion lyin ted
2.406	destructive isis nuclear path wean weapon weaponry weapons
2.424	ahead budged budget budget's budgets schedule scheduled schedules
2.427	angel angela angelic angels evangelical evangelicals evangels religion
2.462	border borderline border's borders bore open secure southern
2.493	factories factory jobs lost manufacture manufactured manufacturer manufacturers manufactures manufacturing nafta
2.570	appoint appointed appointing appointment appointments appoints court disappoint disappointing disappointment disappoints judge judges justices point reappointed scalia supreme uphold
2.638	air carrie carried carrier carries condi condition conditional conditioned conditioner conditioners conditioning conditions fire indianapolis preconditions unconditionally unit
2.771	destabilize destabilized destabilizing east ira iran iraq libya middle rack stab stability stabilize

Table 1: Selected clusters from our dataset. Mean of squared errors (MSE) is presented for each cluster as well as the full accounting of cluster words.

max sizes of 100, 3, and 6 respectively. These vectors were then clustered using spherical *k*-means[4] with a *k* of 300.

The chart was arranged horizontally for easy visual inspection of the resulting cluster time-series relationships. Several findings stood out. Political topics which were closely related to a specific event were successfully captured by our time-series chart and are represented by large and sudden surges in frequency.

For example, there is a large spike in mid November 2015 for a cluster summarized by *Syrian refugee refugees Syria's*. This appears to correspond to the terror attacks in Paris on November 13, 2015 and the U.S. political rhetoric that followed. Another example is that of a cluster representing the word *apologizing*, which had a huge spike in October. This corresponds to a political scandal surrounding our subject in early October for which they apologized frequently thereafter.

In addition to sudden spikes and anomalies, long term trends concerning broader concepts also presented themselves in our charts. A cluster represented by the terms *billion, deficit, dollars,* and *millions*, is found frequently early in the campaign, but subsequently declines, presumably in favor of other talking points.

The impact of the representation of words as summed n-gram embeddings is seen in many clusters. A notable example is one we labeled the *Sergeant Bergdahl* cluster, which contained several nonsensical near-spellings of the surname: *bergdorf, berger*, and *dahl*. At the same time, this grouping also contained the semantically appropriate words *dirty*, *rotten*, and *trader* (which, presumably, is a homophone for traitor). Another example of this homonymous/semantically related duality is a cluster containing the terms *Dr Ben Carson* (separately), the rough homophones *larson* and *parson*, but also the relevant topical words *race* and *races*. This fixation on homonyms was a detriment to the semantic cohesion of some clusters, but also improved the results of others. Large groups of words representing abstract concepts like actions, where all words ended in *-ing* and many were similar in spelling to *running*, were uncovered by our analysis. We provide a selection of discovered clusters in Table 1.

6.1 Evaluation

Systematic empirical evaluation was performed to measure the effects of each component compared to a LDA topic model. Models were trained for combinations of word2vec and fastText embeddings clustered by both *k*-means and spherical *k*-means. Models were trained over a range of $100 \leq k \leq 1000$ with a step of 5 clusters/topics.

Pointwise mutual information (PMI)[22, 23] over a sliding window of 10 words[21] was chosen as our evaluation method. This was selected because other methods commonly used were found to be unsuitable. External evaluation[20, 21] and WordNet[10] based techniques[13, 23] suffered due to the high number of misspelled and mistranslated terms in our dataset. Document similarity [20, 22] methods were not used because individual speeches had large amounts of thematic overlap and are generally very similar. This left us with co-occurrence methods—namely windowed PMI[21]—which encapsulates our intent to capture tightly coupled small groups of words which characterize specific references.

Relative cluster instability and coherence were the focus of our empirical analysis. Word2vec-based models tended to be highly unstable, regardless of *k*-means variant, yet they yielded high PMI scores with the right choice of *k*. The PMI metric identified word2vec-based model sensitivity to specific number of topics. This is observed as the diverging trends in word2vec PMI scores in our evaluation graph in Figure 4.

FastText clusters, in general, were much more stable than their word2vec counterparts, with spherical fastText clustering exhibiting nearly the amount of stability as LDA topics. Signs of slight instability only appeared towards the upper end of our analysis ($k \gtrapprox 900$). LDA produced topics which scored second-to-highest in low numbers of *k*. No significant signs of instability were displayed with LDA, even through 1000 topics.

To investigate the diverging scores, we tracked a cluster that appeared across all embedding models: the *carrier air* cluster, which references Carrier Corporation's air conditioner manufacturing operation in Indianapolis and subsequent layoffs. This cluster was

[4] https://github.com/clara-labs/spherecluster

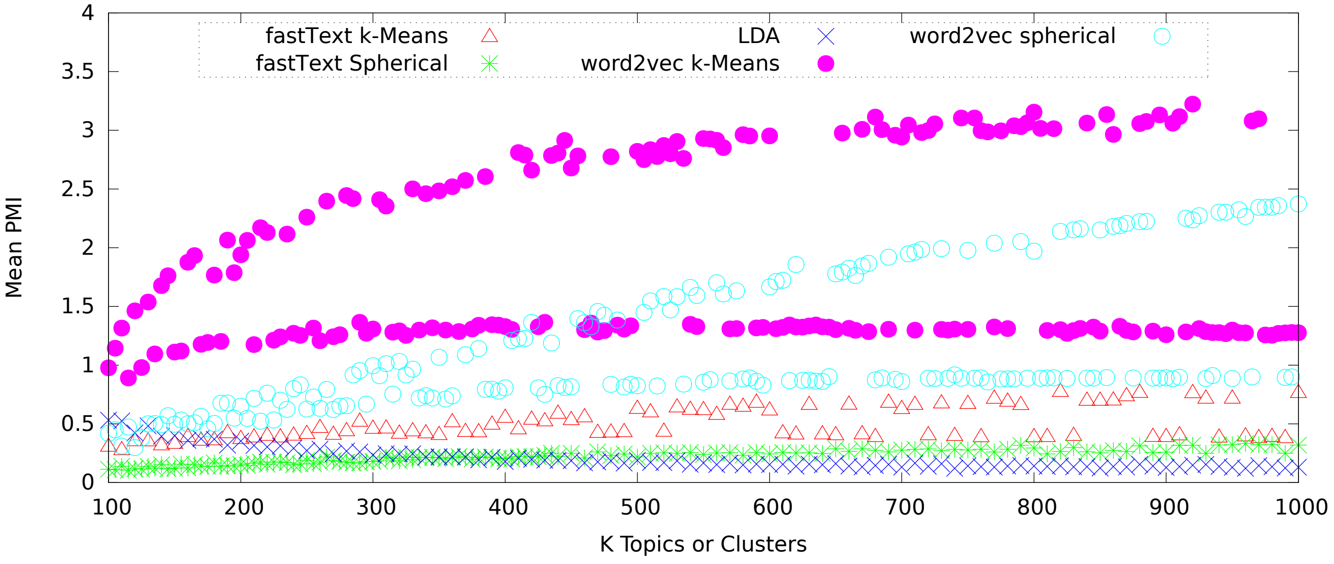

Figure 4: Mean pointwise mutual information (PMI) for several variations of clustered embeddings vs LDA over a range of k clusters/topics.

identified by the terms *carrier* and *air*. The highest-scoring clusters also often contained the terms *Indianapolis*, *fired*, *unit* and many erroneous variations thereof. This highly-specific word group was absent from LDA model outputs, outside of large, incoherent sets.

In spherical fastText clusters, the most stable among the models, *carrier air* clusters consistently appeared between $200 \lessapprox k \lessapprox 800$. These had a total set union of the terms *air, carrier, conditioned, conditioner, conditioners, conditioning*.

Depending on the specific number of clusters selected, k-means word2vec clusters fluctuated between small referential topics and big nonsensical ones. A survey of word2vec k-means models between a k of 705 and 1000 exposed small, topical clusters like *air carrier conditioner conditioners conditioning unit* (PMI of 4.59). With slight changes of k, clusters consisting of mostly random terms (PMI of 0.04 or less) were found. This pattern explains the divergent nature of the average word2vec PMI scores and illustrates the relative importance of selection of k with such models.

The fastText clustering models incorporated terms which represented misspellings of otherwise correct words into single clusters. This resulted in clusters which ultimately represented single words, but in many misspelled variations. Since single-item clusters tend to rarely co-occur, this meant that fastText clusters had generally low PMI scores.

The dual nature of clustering typographically incorrect words into semantically coherent groups is both a syntactic and semantic task which is difficult to empirically quantify using common evaluation methods for topic models. Further exploration of this would be useful in identifying ways to automatically optimize the embedding strategy and choice of k in the extraction of specific references from speech transcriptions.

7 CONCLUSIONS

Despite the high noise and error filled nature of our real-world data, we found that it is possible to discover both abstract concepts and sparse event references. Further, our technique also extracts relevant homonyms and incorrect transcriptions, aided greatly by dense subword embeddings and an extension of classic clustering techniques. Our method is unsupervised and uses minimal preprocessing. It should be possible to extend this framework to other similarly corrupted datasets like documents scanned with optical character recognition[2].

As a time-series analysis technique, we have shown that it is simple to discover, count and perform trend analyses on both general topics and highly specific references. Inspection and discovery of interesting trends is intuitive and can be performed without the help of domain experts. The use of indicators, such as an exponential moving average[9] or Kalman filter[2], could be used for the automatic mining of topical movements.

The nature of word embeddings is still not well understood, but extensions of work in this area, particularly clustering[12], is promising for many fields, including journalism and political science. Future improvements in these methods could be generalized to count the frequencies specific relationships, such as the number of times an entity is mentioned in the transcripts of a political debate or the number of references to specific organizations in the closed-captions of local-access government television.

REFERENCES

[1] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research* 6, Sep (2005), 1345–1382.

[2] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 113–120.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606* (2016).

[5] Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models.. In *Nips*, Vol. 31. 1–9.

[6] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.

[7] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, Vol. 6. Genoa, 449–454.

[8] Inderjit S Dhillon and Dharmendra S Modha. 2001. Concept decompositions for large sparse text data using clustering. *Machine learning* 42, 1 (2001), 143–175.

[9] Brian Dickinson and Wei Hu. 2015. Sentiment analysis of investor opinions on twitter. *Social Networking* 4, 03 (2015), 62.

[10] Christiane Fellbaum. 1998. *WordNet.* Wiley Online Library.

[11] Justin Grimmer. 2010. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases. *Political Analysis* (2010), 1–35.

[12] Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting Embedding Features for Simple Semi-supervised Learning.. In *EMNLP*. 110–120.

[13] Graeme Hirst, David St-Onge, et al. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database* 305 (1998), 305–332.

[14] Sergei Koltcov, Olessia Koltsova, and Sergey Nikolenko. 2014. Latent dirichlet allocation: stability and applications to studies of user-generated content. In *Proceedings of the 2014 ACM conference on Web science*. ACM, 161–165.

[15] Lingpeng Kong, Chris Alberti, Daniel Andor, Ivan Bogatyy, and David Weiss. 2017. Dragnn: A transition-based framework for dynamically connected neural networks. *arXiv preprint arXiv:1703.04474* (2017).

[16] Hank Liao, Erik McDermott, and Andrew Senior. 2013. Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 368–373.

[17] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings.. In *AAAI*. 2418–2424.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[20] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 262–272.

[21] David Newman, Sarvnaz Karimi, and Lawrence Cavedon. 2009. External evaluation of topic models. In *in Australasian Doc. Comp. Symp., 2009*. Citeseer.

[22] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 100–108.

[23] Pavel Pecina. 2008. A machine learning approach to multiword expression extraction. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, Vol. 2008. 54–61.

[24] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 1105–1112.

[25] Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 424–433.