CS230

Generating news tips for local journalism with topic modeling

Amy DiPierro*
Department of Computer Science
Stanford University
dipierro@stanford.edu

Abstract

Natural language processing and text-mining are rarely used in the field of journalism, where reviewing voluminous corpora by hand is the norm. We test the power of topic modeling algorithms to discover newsworthy themes in a corpus of more than 3,000 meeting agendas and meeting minutes collected from government agencies in the Bay Area. We find these methods successfully detect certain emergent news events and suggest additional steps to reduce the number of illegible or irrelevant topics surfaced by the models.

1 Introduction

News reporters were once fixtures of local government meetings around the United States, closely monitoring city councils and other agencies on behalf of the public. But more than 2,000 newspapers have shuttered over the past 15 years.[1] Journalists that remain are spread too thin to attend every meeting. The consequence is that news stories of civic importance go undetected and unreported.

To address this problem, Stanford University students working on a project called Agenda Watch have scraped more than 3,000 meeting-related documents posted by Bay Area governments. These documents include both agendas that outline the items to be discussed at upcoming meetings as well as minutes that record what was discussed at previous meetings.

We seek to mine this corpus to surface newsworthy topics for local journalists to investigate. We present two algorithms for this topic modeling task. Online Latent Dirichlet Allocation [2] inputs preprocessed excerpts of meeting agendas and minutes and outputs predicted topics as well as coherence scores for those documents. The generative topic embedding model TopicVec [3] inputs these same excerpts of documents as well as pre-trained word embeddings and unigram probabilities extracted from Wikipedia. Similarly, it outputs predicted topics, coherence scores and topic vectors.

While text-mining methods have been adopted in much of academia and industry, they are still novel in the field of journalism.[4] Today, reporters review documents like those in our corpus almost exclusively by hand, a tedious and time-consuming task. We demonstrate here that topic modeling can augment the capabilities of journalists by identifying key themes in voluminous government documents.

*The author is an M.A. candidate in Journalism. Her work has appeared in USA TODAY and the AP.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

Latent Dirichlet Allocation (LDA), a Bayesian probabilistic topic model developed in Blei [6], seeks to learn the latent themes of a corpus by representing documents as mixtures over topics, and topics as distributions over words. It is often summarized as a two-step generative process. In the first step, for every topic k, topic $_k$ $Dir(\eta)$ is drawn. In the second step, for a document d : (a) a topic distribution θ_d $Dir(\alpha)$ is chosen and (b) for each word in a document, a topic and word are chosen according to categorical distributions.

LDA thus groups words into topics based on their collocation across documents, a process that yields coherent topics when applied to a sufficiently large corpus. While LDA has proven to be a powerful way to discover useful structure in unstructured corpora, the model can be computationally expensive for extremely large or streaming corpora and fails to capture changes in topics over time. [7] and [8] modify LDA to account for the latter limitation, proposing two methods to analyze the evolution of topics through time. [2] addresses the former weakness, using online stochastic optimization with a natural gradient step to reduce the computational time needed to compute LDA. This method, online variational inference for LDA, or Online LDA, improves speed by taking as input chunks of the corpus and updating the LDA model after each chunk rather than processing the entire corpus in one pass as in LDA.

More recent work attempts to integrate LDA with advances in using dense word vector representations to encode meaningful relationships between words. Das et. al. [9] introduce Gaussian LDA, which uses continuous space word embeddings drawn from a multivariate Gaussian distribution, causing the model to group words already known to be semantically related into topics. [9] overcomes the out of vocabulary limitations of traditional LDA. Moody [10] proposes *lda2vec*, a model that learns both word vectors and, at the document level, Dirichlet-distributed latent mixtures of topic vectors. It leverages the interpretability of topics generated by LDA while integrating the Skipgram Negative-Sampling approach developed in [11] to train word embeddings. The merits of this approach are uncertain; it has neither been tested against LDA nor word2vec baselines. Li et. al. [3], discussed in further detail below, similarly combine topic modeling with word embeddings.

Because human evaluation of topics is time-intensive, coherence measures have emerged as one way to automatically measure the understandability of topics generated by unsupervised methods. In [12], several coherence metrics are compared and evaluated based on their correlation with human judgment. Nikolenko [13] proposes new coherence metrics using distributed word representations, finding that these measures more closely track human judgment.

3 Data and Features

Our data consists of 3,200 agendas and minutes corresponding to meetings to be held by 16 local government agencies in the Bay Area between December 2019 and December 2020. We preprocess these texts in a pipeline designed to isolate distinct meeting items and to remove "boilerplate" text repeated across meeting documents, such as standard disclosures regarding meeting procedures and public access laws. We first compute the "diff" between each incoming document and an example document from the same government agency; when the documents share identical text passages, we consider this text to be boilerplate and delete it from the document. We then split each incoming document into chunks of at least 200 words, using word count as a heuristic to isolate distinct agenda items. This yields roughly 8,600 texts treated by the model as distinct documents.

We next apply standard preprocessing methods, tokenizing the text, extracting bigrams, removing stop words and applying a part of speech filter in order to keep only nouns, adjectives, verbs and adverbs. Because we want our model to recognize topics across government agencies rather than within one agency, we additionally use entity extraction to remove the names of people mentioned in the documents.[5] This prevents the model from learning the names of, e.g, city council members as a distinct topic. By the same logic, we define a list of custom stop words that includes the names of places covered by the corpus.

We split our data into a training set of 6,897 meeting document excerpts and evaluate topic legibility and coherence on a dev set of 862 unseen excerpts. We conduct final testing on a set of 863 additional unseen documents.

4 Methods

4.1 Online Latent Dirichlet Allocation

For our use case, the ability to train and update a model with a stream of new incoming documents is imperative as our corpus grows. We thus apply the Gensim [14] implementation of Online LDA to our corpus of meeting documents because of its speed advantage over traditional LDA.

Online LDA approximates the true posterior distribution of topics β, topic proportions θ and per-word topic assignments z using a variational distribution; z is parameterized by ϕ, θ is parameterized by θ and β is parameterized by λ. When these parameters maximize the Evidence Lower BOund (ELBO), it is equivalent to minimizing the distance between the true posterior distribution and the variational distribution as measured by Kullback-Leibler divergence. In Online LDA, we reach this maximum by using a stochastic natural gradient algorithm that initializes λ randomly, and iteratively updates it with a weighted average of its previous values until convergence.

4.2 TopicVec

We next apply TopicVec. Given a corpus of documents, word embeddings and unigram probabilities as inputs, the model outputs topic embeddings in the same embedding space as words. The model's core training strategy is a variational inference algorithm. In its first step, the algorithm disregards topics to obtain optimal embeddings and bigram residuals. In the second step, these embeddings and bigram residuals are used to find the optimal topic embeddings. This is accomplished by maximizing the variational free energy $L(q, T)$ with a Generalized Expectation-Maximization algorithm given by

$$L(q,T) = \sum_{i=1}^{M}\{\sum_{k=1}^{K}\sum_{j=1}^{L_i}(\pi_{ij}^{k} + \alpha_k - 1)(\psi(\theta_{ik}) - \psi(\theta_{i0}))+ Tr(T_i^T \sum_{j=1}^{L_i}\pi_{ij}^T) + r_i^T \sum_{j=1}^{L_i}\pi_{ij}\} + H(q) + C_1,$$

where α_k is a Dirichlet parameter for a topic, T_i is the topic matrix of document i, v_{s_i} is the embedding of word s_i, r_i is a vector constructed by concatenating all topic residuals, $H(q)$ is the entropy of q and C_i is a constant and the variational distribution is used is $q(Z, \phi; \pi, \theta)$.

5 Experiments

For both Online LDA and TopicVec, we first choose the number of topics K based on a qualitative assessment of topic legibility. We then tune each model to optimize coherence given K while still satisfying legibility and convergence constraints. We measure coherence with *UMass coherence* [14], a common benchmark metric, as well as a coherence metric derived using pretrained GloVe vectors, following [13].

5.1 Online Latent Dirichlet Allocation

We first vary K between 15 and 300, assessing the quality of topics manually after each run. We choose $K = 100$ and then tune the variables *decay*, a weight determining what percentage of the previous value of the variational parameters λ to "forget" with each new document; *chunksize*, the number of documents in each training chunk; and *offset*, a hyperparameter controlling how much to decelerate during the initial iterations. We learn the asymmetric prior η from the data using Gensim's 'auto' function and define a fixed normalized asymmetric prior of $\frac{1}{100}$. Although not strictly hyperparameters, we also "tune" our model by varying the thresholds at which we discard tokens used frequently (defined by the variable *no_above* in Gensim) or infrequently (no_below in Gensim).

We seek to adjust all values to maximize *GloVe coherence* and obtain *UMass coherence* approaching zero, given the qualitative constraint that topics produced should be human legible and the quantitative constraint that topic composition must converge to a relatively stable point. We vary *chunksize* between 1 and 16,384, *offset* between 1 and 1,024 and *decay* between 0.5 and 1, following the ranges

in [2]. For number of passes, we explore values between 1 and 500. Finally, we adjust *no_above* between 0.1 and 1 and *no_below* between 1 and 25.

5.2 TopicVec

In addition to our training data, we input a 500-dimension word embeddings vector file and a text file containing one-grams used in the original TopicVec paper into the model. We first experiment with the number of topics, running trials with K = 18, 23, 50 and 100 topics. After inspecting the results, we find $K = 100$ yields the most legible topics. Holding K steady, we conduct a hyperparameter search for δ, the initial learning rate, α_0, the Dirichlet hyperparameter on the null topic and α_1, the Dirichlet hyperparameter for all other topics. For each of these inputs, we sample a range of values between 0.05 and 0.15, confining our search to a window near the values used in the original paper. We perform 100 GEM iterations, following Li [3].

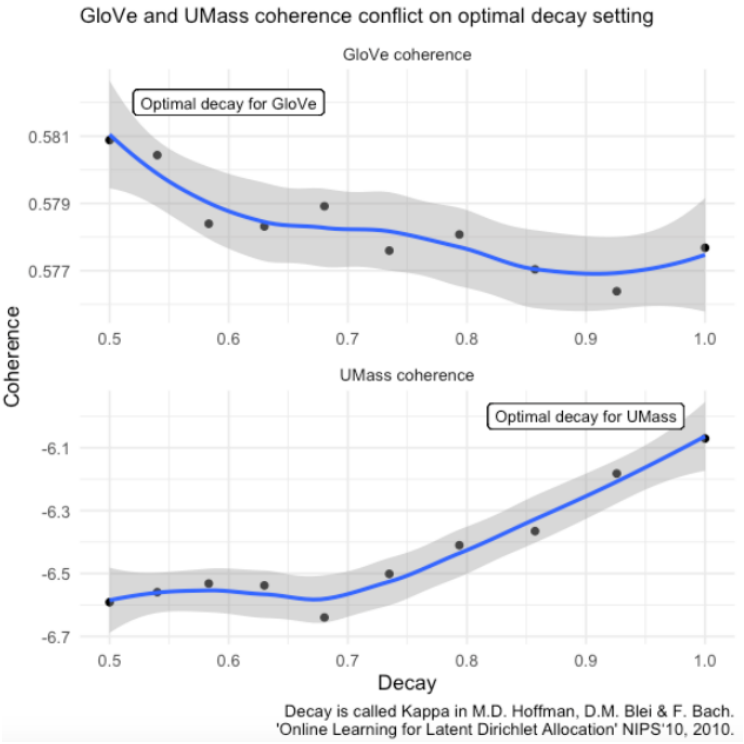

Figure 1: Comparing *GloVe coherence* and *UMass coherence* for the hyperparameter *decay*.

6 Discussion

For the TopicVec model, we find that the model requires little tuning; the original values of α_0, α_1 and δ from [3] are appropriate for our use case and appear to generate the best quality topics.

For Online LDA, however, we find that deviating from the benchmark inputs in [2] is advantageous. We also find that aggressive word occurrence filtering improves topic legibility. Full training and dev results for both models are available on GitHub[2]; the inputs selected for testing as well as test results for both models are summarized in Tables 1 and 2.

Qualitatively, both models pass a basic sanity check, successfully identifying general *themes* in the underlying corpus, including real estate development and energy. The models also meet with some success in isolating specific *trends* and *events*. Online LDA, for example, correctly identifies a topic describing the wave of eviction

TABLE 1: Test settings

Online LDA		TopicVec	
Input	Value	Input	Value
num_topics	100	K	100
decay	0.05	alpha0	0.1
offset	384	alpha1	0.1
chunksize	1024	iniDelta	0.1
iterations	1000	iterations	100
passes	500	-	-
no_above	0.143	-	-
no_below	25	-	-

[2]https://github.com/DiPierro/cs-230-project

TABLE 2: Test results

Model	Online LDA	TopicVec
GloVe Coherence	0.59	0.506
Umass coherence	-6.16	-8.56
Novel topics	**Evictions:** [rent, tenants, residential, residents, pandemic, hotel, evictions, fees, urgency, rooms]	**Police:** [fierce, bullets, policing, justifying, crowds, incident, lethal, gun, harm, force]
	Environment: [mitigation, cleanup, level, shoreline, costs, bay, legacy, closure, rise, lagoons]	**Protest:** [batons, protester, misusing, bullets, altercation, rubber, incident, abductors, projectiles]
Shared topics	**Energy:** [gas, electric, emissions, codes, reach, response, greenhouse, buildings, natural, reduce]	**Energy:** [emissions, gas, reach, electric, carbon, codes, natural, greenhouse, appliance, cooking]
	Development: [residential, determination, zoning, appeals, units, response, exemption, findings, scheduled, family]	**Development:** [neighborhood, feet, zoning, design, square, pedestrian, downtown, located, commercial, residential]
	Election: [election, measure, november, ballot, november_election, ballot_measure, amendments, voters]	**Election:** [ending, election, seat, appointment, alternate, nomination, unexpired, november, ballot]

moratoria passed in the Bay Area over recent months[3]; TopicVec forms two topics discussing different aspects of protests against the police and policing itself. Both models detect discussions on the November 2020 election. Each of these topics would likely be of interest to local journalists.

However, while the results above are promising, many of the topics generated by both models do not usefully summarize the meeting documents. For example, both models learn topics obviously drawn from "boilerplate" passages, such as topics summarizing the standard advisory before the opening of a public comment period. They also generate a handful of nonsensical topics.

Additionally, we find that coherence measures did not consistently track human judgment of legibility, making it difficult to tune the model automatically according to this metric. In both models, experiments using fewer topics yielded higher coherence scores, yet hand inspection of these results indicated that these topics were either illegible or filled with irrelevant meeting jargon rather than the substance of meeting discussion. In Online LDA, coherence was highest during experiments with only one pass, or epoch, through the data. However, in these experiments, the model failed to converge to a set of stable topics, and hand inspection of the topics again showed inferior legibility relative to the topics generated by experiments with more passes through the training data.

While *GloVe coherence* and *UMass coherence* often tracked one another, on several occasions the two measures diverged, with each metric indicating a different hyperparameter setting to yield the most coherent topics. This pattern is evident in Figure 1. In these instances, we typically tuned the hyperparameter after qualitatively assessing topics generated by each setting.

7 Future work

We present here two methods for modeling topics from a collection of local government meeting minutes and agendas using the algorithms Online LDA and TopicVec. Our results suggest that applying topic modeling to this corpus can automatically generate news tips for local journalists to screen and, potentially, investigate with further reporting. We see opportunities within and beyond the topic modeling domain to refine our work further.

A logical next step is to introduce the variable of time. One approach is to modify our current topic models to run on time slices of documents and to track changes in topic composition after each time slice. Dynamic LDA [7] or the method described in [8] provide two viable alternatives.

We also see opportunities to recast our news-finding objective in terms of natural language processing tasks beyond topic modeling. Following [16], we can conceive of the problem of identifying news stories in our corpus as an event detection task. Another promising approach is to apply methods such as Paragraph Vector [17] to our corpus with the goal of finding meeting document excerpts close to one another in vector space.

We believe more focused data preparation has the greatest potential to improve signal detection from our unstructured corpus, regardless of whether this signal is represented as a trending topic, an emergent news event or a cluster of similar documents. Thus, a third avenue for further research is to train models to aid us in preprocessing. For example, we could generate synthetic meeting documents and use them to train a model that extracts individual agenda items from meeting documents.

[3]See L.D. Sault. "Bay Area eviction bans", *The Mercury News*, 2020 https://bayareane.ws/3nmDDnA

8 Acknowledgements

Agenda Watch began as a project in Fall 2019's Exploring Computational Journalism course. The corpus of documents we analyze was scraped by ECJ student Christopher Stock.

9 References

[1] P. M. Abernathy. "News Deserts and Ghost Newspapers: Will Local News Survive?" UNC Hussman School of Journalism and Media, 2020.

[2] M. D. Hoffman, D. M. Blei & F. Bach. "Online Learning for Latent Dirichlet Allocation." In *Advances in Neural Information Processing Systems 23 (NIPS)*, 2010.

[3] S. Li, T. Chua, J. Zhu & C. Miao. "Generative Topic Embedding: a Continuous Representation of Documents." In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[4] J. Stray. "Making Artificial Intelligence Work for Investigative Journalism." Digital Journalism, 2019.

[5] M. Honnibal & i. Montani. "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.

[6] D. M. Blei, A. Y. Ng M.I. Jordan. "Latent dirichlet allocation." *The Journal of Machine Learning Research*, March 2003. https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

[7] Blei, David M. Lafferty, John D. "Dynamic Topic Models." In *Proc. of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[8] Hall, David et al. "Studying the History of Ideas Using Topic Models." In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2008.

[9] R. Das, M. Zaheer, C. Dyer. "Gaussian LDA for Topic Models with Word Embeddings." In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015.

[10] C. E. Moody. "Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec." *arXiv preprint arXiv:1605.02019*, 2016.

[11] T. Mikolev, I. Sutskever et al. "Distributed Representations of Words and Phrases and their Compositionality." In *Proceedings of NIPS*, 2013.

[12] M. Roder, A. Both & A. Hinneburg. "Exploring the space of topic coherence measures." In *WSDM '15*, 2015.

[13] S. I. Nikolenko. "Topic Quality Metrics Based on Distributed Word Representations." in *SIGIR '16*, 2016.

[14] R. Řehůřek & P. Sojka. "Software Framework for Topic Modelling with Large Corpora." In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010.

[15] D. Mimno, H. M. Wallach, E. Talley, M. Leenders & A. McCallum. "Optimizing semantic coherence in topic models." In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, 2011.

[16] X. Liu, A. Nourbakhsh, Q. Li et al. "Reuters Tracer: Toward automated news production using large scale social media data." In *2017 IEEE International Conference on Big Data (Big Data)*, 2017.

[17] Q. Le & T. Mikolov. "Distributed Representations of Sentences and Documents." In *Proc. Of the 31st International Conference on Machine Learning*, 2014.