A FRAMEWORK FOR UNDERSTANDING THE IMPORTANCE OF REFERENCE TERM VECTOR PREPARATION IN LATENT DIRICHLET ALLOCATION

Marcus Hilliard, Rikita Hirpara, Doron Meizlik, Katie Zink

Northwestern University

Dr. Alianna J. Maren

March 14, 2021

Abstract

This research focuses on advancing the understanding of what truly drives Latent Dirichlet Allocation (LDA) model performance, tracing back to reference term vector preparation and drawing a direct line between that process and the model output. Using a corpus of documents focused on the policy priorities of the Biden Administration and a series of experiments testing incrementally more sophisticated data preparation procedures, we demonstrate not only that topic mapping is incredibly sensitive to even the smallest changes in its input space, but more importantly, propose a theory for understanding the source of that volatility.

Introduction

The purpose of this research is to elucidate the connection between reference term vector preparation and the results of topic mapping. Using a corpus of documents focused on the early days of the Biden Administration and its legislative priorities, we will perform Latent Dirichlet Allocation (LDA) to probabilistically allocate these documents into a fixed number of topics. In doing so, we seek to understand what is truly driving the topic maps and draw a line between the choices made in term representation and the output we observe.

This research consisted of the following steps. Beginning with exploratory data analysis and reading each article, we first manually determined a sense of ground truth as to the content of each article. Then we sought to better understand our corpus in two ways: a bottom-up approach in which we analyzed our documents' words and phrases, and also a top-down approach in which we built an ontology on the world of the Biden Administration's policy priorities to guide our thinking. The Word2Vec algorithm proved instrumental in the former, helping us to see which terms were similar, or at least related; later on this would help us understand how certain terms formed the "glue" between certain topics and documents. Our ontology, on the other hand, pictured below in Figure 1, comes at this same question from a very different perspective but similarly provides its own view as to how the different topics related to one another.

Corpus Ontology for Biden-Harris Administration

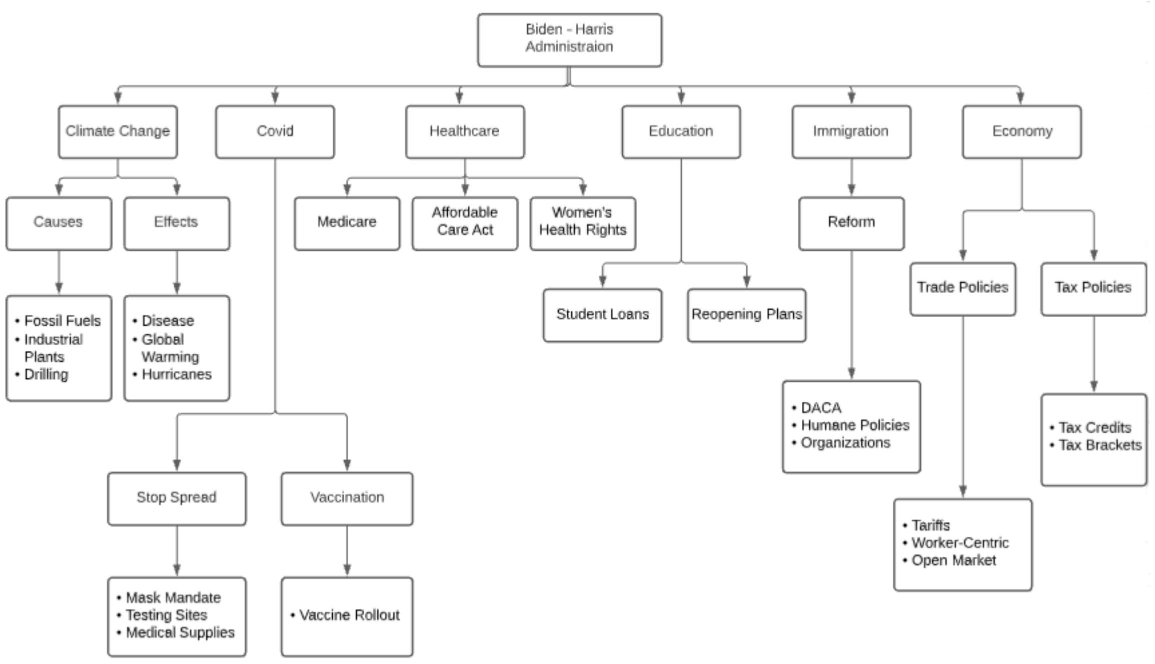

Biden - Harris Administraion
Climate Change
Covid
Healthcare
Education
Immigration
Economy
Causes
Effects
Medicare
Affordable Care Act
Women's Health Rights
Reform
Trade Policies
Tax Policies
Student Loans
Reopening Plans
• Fossil Fuels • Industrial Plants • Drilling
• Disease • Global Warming • Hurricanes
• DACA • Humane Policies • Organizations
• Tax Credits • Tax Brackets
• Tariffs • Worker-Centric • Open Market
Stop Spread
Vaccination
• Mask Mandate • Testing Sites • Medical Supplies
• Vaccine Rollout

Figure 1. This ontology of the Biden-Harris Administration's policy priorities gives a high-level worldview of what matters most to the incoming Administration and will serve as a reference for the research team as it interprets the output of its LDA topic mapping.

With a good sense of the corpus and the terms that comprise it, we moved onto document vectorization and feature engineering. But while we list this step here, this turned out to be a highly iterative step that we returned to over and over again throughout our research as we refined these representations. These representations and the choices we made in developing them would prove to ultimately define our topic maps. We will demonstrate the direct link between data preparation and how well the ultimate algorithmic LDA results conform to our ground truth expectations. The next step in our research process was taking these vector representations and running them through multiple clustering algorithms, including k-means and hierarchical clustering. While we were ultimately after topic maps and not clusters, we used clustering as an intermediate step to help us better understand our document groupings and how our documents relate to one another. This was so that we could ultimately decide on the number of topics we wanted to use in our LDA and define those topics. After deciding on seven topics, we ran our topic mapping LDA to probabilistically group each document into these topic areas: the fight against COVID-19, student loans and other areas for executive actions, healthcare policy beyond the pandemic, climate change, women's and justice issues, economic stimulus and how to pass the overall agenda, and immigration policy and trade policy. Finally, we visualized our results, visualized what led to those results, and iterated our feature engineering steps until we made sense of how the terms in our documents led to those results.

It is this last part that is the real focus of the analysis that follows. While the algorithmic steps that led us to the final model are important, our research aim is to enhance the understanding of the link between the LDA model's input vector representations and the resultant topic mapping. We do this first by demonstrating the sensitivity of the model output to the manual choices made during feature engineering, and then by proposing a framework for explaining that connection.

Literature Review

Natural language processing (NLP) has evolved from primarily rule-based systems to more sophisticated probabilistic systems. More recently, with the rise of powerful computers, we can now train systems that would have been impossible to develop a few years ago. Also, we can now capture more complicated patterns using advanced algorithms to answer associated questions in different applications.

In each subsection below, which generally follows the structure of this study, we provide an overview of some of the tools and algorithms used in this research specifically.

Exploratory Data Analysis

To understand the words in our corpus and their relation to one another, we utilize distributional semantics, which hypothesizes that those words which are distributed the same way in text or speech will have similar meanings. According to the distributional theory, the words *keyboard* and *piano* might occur together more frequently in text because they infer a similar idea. Thus, when words and phrases are projected into a semantic space, those words that have similar meanings are closer together. Those that have less similar meanings are further apart.

Word2Vec (Mikolov 2013) is a complex, multi-level neural network that projects words into a semantic space, which can then be used to determine the semantic distance or semantic proximity. The output is a space filled with vector representations of words. Word vectors that are semantically closer together are more closely related than word vectors that are farther apart. The emphasis of the Word2Vec model is to produce word embeddings that have transformed the high-dimensional, sparse vectors often associated with large corpora into lower-dimensional, dense vectors that are much more computationally friendly. The training of a Word2Vec model does not involve dense matrix multiplication, making the training very efficient. Researchers have tried using Word2Vec to find the meaning of a word in context (Wang 2014), to automatically determine human attitudes in the text (Xue, Fu, and Shaobin 2014), and even to ascertain political ideology (Iyyer et al. 2014).

Feature Engineering

Preprocessing input data is a critical step in NLP because text data must be transformed into a machine-readable format. Much of this research focuses on some of these steps and shows why topic mapping is so sensitive to the decisions made in feature engineering. There are many ways to preprocess data for machine learning. These include but are not limited to: removing whitespace, removing extra characters, converting all words to lowercase, stemming, lemmatization, removing stop words, creating equivalence classes, and creating vector representations. We will later show that the manually-decided user interventions to refine equivalence classes and stop word lists are of particular importance in topic mapping.

The formation of equivalence classes (Manning, Raghavan, and Schütze 2008) is an essential part of preprocessing data for NLP. This method is a way to normalize your data, despite superficial differences, so that machines can better understand it. From our corpus, one example is the many different ways COVID-19 is referred to throughout the various documents. Some of the different ways this might appear are “COVID-19”, “COVID”, “coronavirus”, or “virus”. We can form an equivalence class to change all of those instances to “covid”. This will allow us to better understand the significance of these words in our research.

Lemmatization (Dataquest 2019) is the algorithmic process of considering word tense and usage to determine and reduce each word to its base form. An example would be words like “connect”, “connected”, “connecting”, and “connection”, which are all a little different but have the same essential base meaning. Stemming, on the other, is a much cruder process that has the same nominal goal, but simply chops off prefixes and suffixes without semantic appreciation to reduce words to their root, or stem. Many researchers have used lemmatization and stemming to study the application of these algorithms in different languages, as well. These specific researchers are looking at stemming in Swedish (Carlberger et al. 2001) and Hindi (Paul et al. 2013).

Document Vectorization

The bag-of-words model in NLP is one of the most popular ways of vectorizing documents (Brownlee 2017), and also happens to be the vectorization of choice for the topic modeling algorithm at the heart of this study. Bag-of-words is used to extract features from text to use for modeling. This model describes, for every word or n-gram in the corpus, a measure of the presence of those words. The model takes these words and essentially throws them into a "bag", losing their order and structure in the process but retaining the knowledge of their frequency. One famous use case for bag-of-words is in training robots to better understand rooms within an apartment (Filliat 2007).

The Term Frequency - Inverse Document Frequency (TF-IDF) approach is used to give a weight to each word in a text document based on how unique it is across the corpus as a whole. This helps to capture the relevancy among words, documents, and categories (Baharudin et al. 2010). TF-IDF is one of the most common approaches for document processing in NLP. This approach allows the researchers to determine which words are significant to the corpus as a whole but aren't overused. Researchers have used TF-IDF to classify hate speech on Twitter (Ayo et al. 2021) and to improve recall and precision in terms of text classification compared to the traditional approach (Yun-tao, Ling, and Yong-cheng 2005).

Doc2Vec (Mikolov and Le 2014) is an extension of Word2Vec and was created by the same individuals from Google, Quoc Le, and Tomas Mikolov. It is an unsupervised neural network that represents documents numerically to find important words and surrounding words within a document. It uses paragraph vectors to look at full sentences, paragraphs, and documents. Doc2Vec was created because the authors felt that other document vectorizers lost the order of words and ignored the semantics of words. Researchers have used Doc2Vec for sentiment classification using a dataset from IMDB (Mikolov and Le 2014) and to find duplicate questions being asked on a forum (Baldwin and Hau 2016). In this particular experiment, Doc2Vec outperformed Word2Vec on all but one subforum.

Clustering

Hierarchical clustering (Yildirim 2020) is a model that creates trees of clusters, or dendrograms, that iterates through a model to group and separate data points. There are two types of clustering: agglomerative clustering and divisive clustering. Agglomerative clustering takes each data point and starts by assuming it is its own cluster. Then similar clusters are combined through a bottom-up approach. Divisive clustering is the opposite of agglomerative clustering and starts with one cluster of all the data points. Then the data is iterated through and separated into different clusters. However, in practice, divisive clustering is rarely used. Currently, researchers are proposing a new hierarchical clustering algorithm (Naumov, Yaroslavtsev, and Avdiukhin 2020) in the hopes of improving the percentage of objectives. Another group of researchers suggests the use of hierarchical clustering to produce lexicons and analyze the sensory attributes of products in the nonstandardized text (Hamilton and Lahne 2020).

K-means clustering is another algorithm for grouping similar data points together to discover patterns, supervised by the desired number of clusters, or k. Within NLP, the algorithm uses the mean of the vectorized data to determine which documents appear in which cluster. Researchers have used the global k-means clustering algorithm that doesn't depend on initial parameter values and uses the algorithm as a local search procedure (Likas, Vlassis, and Verbeek 2003). Researchers have also used k-means clustering to understand the unique sleep schedules of youth athletes if they were multi-sport or individual-sport athletes (Suppiah et al. 2020).

Topic Mapping

Latent Dirichlet Allocation (LDA) is a form of topic modeling and one of the most popular ways to research topic maps. It is also the focus of this research. The algorithm is a generative model for collections of discrete data. This is useful for NLP and text corpora because it allows the researcher to find underlying themes across a corpus (Kim et al. 2019). Once LDA has been trained you get two outputs: term distribution per topic and topic distribution per document. Both will be part of the analysis that follows. Researchers have used LDA as a spam filter (Bíró, Szabó, and Benczúr 2008) and to improve the process of document classification using labels, even though LDA is unsupervised and doesn't traditionally use labels (Wang, Thint, and Al-Rubaie 2013).

There can also be a great success in NLP when using multiple algorithms together. A paper from 2018 (Kim et al. 2019), proposed using multi-co-training (MCT) for document classification using various document representations. The authors used TF-IDF, LDA, and Doc2Vec to take unstructured text and make it into numeric vectors. They found that MCT can achieve better results when using these three methods instead of traditional methods. While we use different algorithms for different purposes in our work, they all serve to contribute to our understanding of how the primary LDA topic models are working.

Data

The corpus of documents that were the focus of this analysis concerns the Biden Administration and its legislative priorities. In total, the corpus contains 53 documents ranging on topics from Biden's climate change agenda, to his COVID-19 strategy, to his immigration plan.

The 53 documents were submitted by 18 unique researchers as part of an effort to compile a thematic corpus, and they vary in length from 166 to 2032 words. The mean article length was 619 words and the median was 457 words. All but three of the documents were published in the two weeks between January 12, 2021, and January 25, 2021, and they come from a variety of news sources spanning the political news ecosystem, with a full 10 articles sourced to the New York Times and another 4 to the Wall Street Journal. The 10 documents submitted to the corpus by the four authors of this paper, for example, focused on the topics of climate change, economic recovery, legislative wrangling, and changes to the Washington political economy.

Before conducting an analysis, though, we had to pre-process our data, relying heavily on Python and its NLTK library for these steps. Beginning with our document corpus, we used NLTK's tokenizer to separate all of the words in each document into separate tokens, remove all punctuation and special characters, expand contractions, remove all non-alphabetic characters, and make all letters lowercase.

However, there were also several feature engineering steps–including deciding on word length criteria, stop words removal, stemming and lemmatization, and the formation of equivalence classes–that would become a foundational part of the analysis itself. A discussion of these key data preparation steps, and, more importantly, how they impacted our ultimate results, will follow in the next section.

Research Design and Modeling Methods

The experimentation in this research that served as the basis for our analysis into the connection between the representation of LDA's input space and model performance can be found in the Feature Engineering sub-section below. However, the remainder of this section reviews the methods by which we would come to understand and interpret our data.

Exploratory Data Analysis

Before conducting our analysis, it was critical to understand our documents and the terms that comprised them. Two algorithms, in particular, helped guide this process: TF-IDF and Word2Vec.

The first algorithm the research team used to investigate the terms in our documents was TF-IDF. This algorithm scores every term in each of the documents by considering both how frequently a term appears in each document and also how uncommon that word is in documents across the corpus; for a term to receive a high TF-IDF score in a given document, it needs to both appear with some frequency in that document and also not be too common throughout the corpus so as to make its appearance in this given document insignificant. After compiling a term list consisting of all unigrams in our corpus documents, we scored each term using the TF-IDF vectorizer in Python's sklearn library. We used this algorithm to get a sense of which terms were important in each document; the highest scoring terms are the ones that ought to drive a given document's topic mapping.

The other algorithm that was essential for understanding the words in our documents and their relation to one another was Word2Vec. Unlike the TF-IDF algorithm, this is not learned based on term and document frequency, but rather is trained using word proximity and contextual similarity to represent words that have similar meanings with similar n-dimensional vectors. This data exploration step is one we returned to throughout our experiment. After training 64-dimension word representations for every word in the corpus, we performed dimensionality reduction using t-SNE to be able to visually represent these vectors on two-dimensional scatterplots. Later, when creating topic mappings with LDA, we visualized the terms in this way, color-coding the dots representing each term by their associated dominant topic, as shown in Figure 2 below.

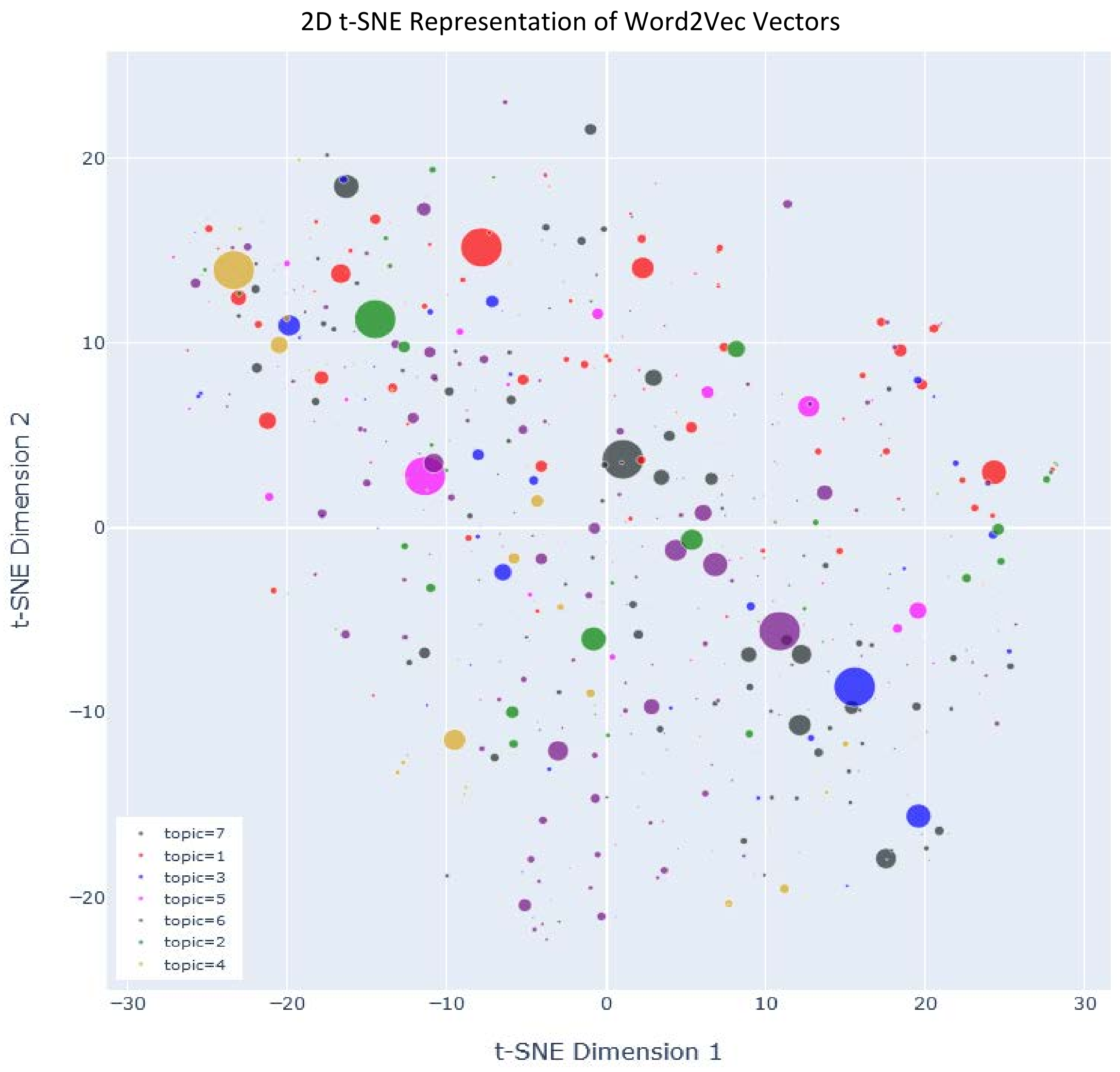


Figure 2. This two-dimensional representation of the Word2Vec vectors for each of the words that were the most dominant in the term distribution for each topic in our final LDA model, with the more important words represented by larger dots, shows how the most important words to the model are distant in two-dimensional space.

Feature Engineering

As mentioned above, four main feature engineering decisions would become our primary experimental parameters: word length criteria, stop words removal, stemming and lemmatization, and the formation of equivalence classes as shown below in Table 1.

We began with a baseline set of data preparation steps that included removing punctuation and special characters, expanding contractions, removing all non-alphabetic characters, and making all letters lowercase. The topic maps that resulted from just this set of preparatory steps would become our baseline model, or Model 0.

Then we performed a series of experiments varying the minimum word length, testing out tokens of minimum length 3, 4, and 5. Model 1, reported below, takes all the baseline data preparation steps and adds in a minimum threshold of four letters in a word.

For our next experiment, we investigated stop word removal. NLTK comes with a default package of stop words, and for Model 2 we incremented on Model 1 by removing all of these default stop words from our corpus as part of our data preparation.

Our next test evaluated the impact of stemming and lemmatization. After investigating the Porter Stemmer, the Snowball Stemmer, and the WordNet Lemmatizer, we decided to use the WordNet Lemmatizer in Model 3, adding it to all the steps used in the previous model.

Following that, we wanted to see how making equivalence classes would or would not improve our results. And so we created a dictionary that maps "covid-19, "covid", "coronavirus", and "virus", for example, all to the same equivalence class. We would create the following equivalence classes, at least at first: "biden", "trump", "democratic", "republican", and "covid". Adding this feature engineering step to all the baseline ones, the minimum word length criteria, stop word removal, and lemmatization made for Model 4.

Finally, we refined our list of stop words to remove and refined our equivalence classes in Model 5. This step took place over numerous iterations and became the final incremental addition to all the previous feature engineering steps. New equivalence classes included:

> "loanforgiveness", "vaccine", "pandemic", "economy", "family", "citizens", "healthcare", and "climatechange".

The additional stop words removed from our final model included:

> "biden", "trump", "president", "federal", "government", "administration", "vice", "cannot", "obama", "presidential", "like", "miss", "year", "much", "kamala", "harris", "department", "white", "director", "say", "says", "said", "saying", "went", and "regardless".

Table 1. Model Development Progression

Model/Experiment Number	*Feature Engineering Steps to Incrementally Add*
Zero (Baseline)	Remove punctuation and special characters, expand contractions, remove non-alphabetic characters, lowercase
One	Apply a minimum word length of 4 letters
Two	Remove NLTK's default library of stop words
Three	WordNet lemmatization
Four	Add in equivalence classes
Five	Refine stop words to exclude and refine equivalence classes

Document Vectorization

For each of these experiments, we vectorized each document in two ways. For the intermediate step of clustering analysis, we utilized the TF-IDF vectorization of each document as the input into the clustering algorithms, focusing on unigrams only to keep our vector dimensionality consistent with the bag-of-word vectors. Bag-of-words vectorization, then, would serve as the input into our Latent Dirichlet Allocation (LDA) topic mapping algorithm.

Clustering

Clustering served as an intermediate result in our analysis. While we ultimately measured the results of our experiments by how the feature engineering choices we made impacted the topic mapping output, we also looked at the intermediate result of various clustering algorithms for two purposes. First, calculating the Within-Sum-of-Squares (WSS) scores and the Silhouette Method helped inform our choice to use seven topics in our topic mapping. Second, the clustering visualizations helped us better understand the relationships between the documents in that topic mapping step. Using the TF-IDF vectors of each document as a basis, we drew dendrograms to cluster the documents based on their cosine similarity and also performed k-means clustering, visualizing the results using multi-dimensionality scaling. An example of one such multi-dimensionality scaling visualization, using k-means clustering on the TF-IDF vectors prepared in Experiment 5, with k = 7, appears in Figure 3.

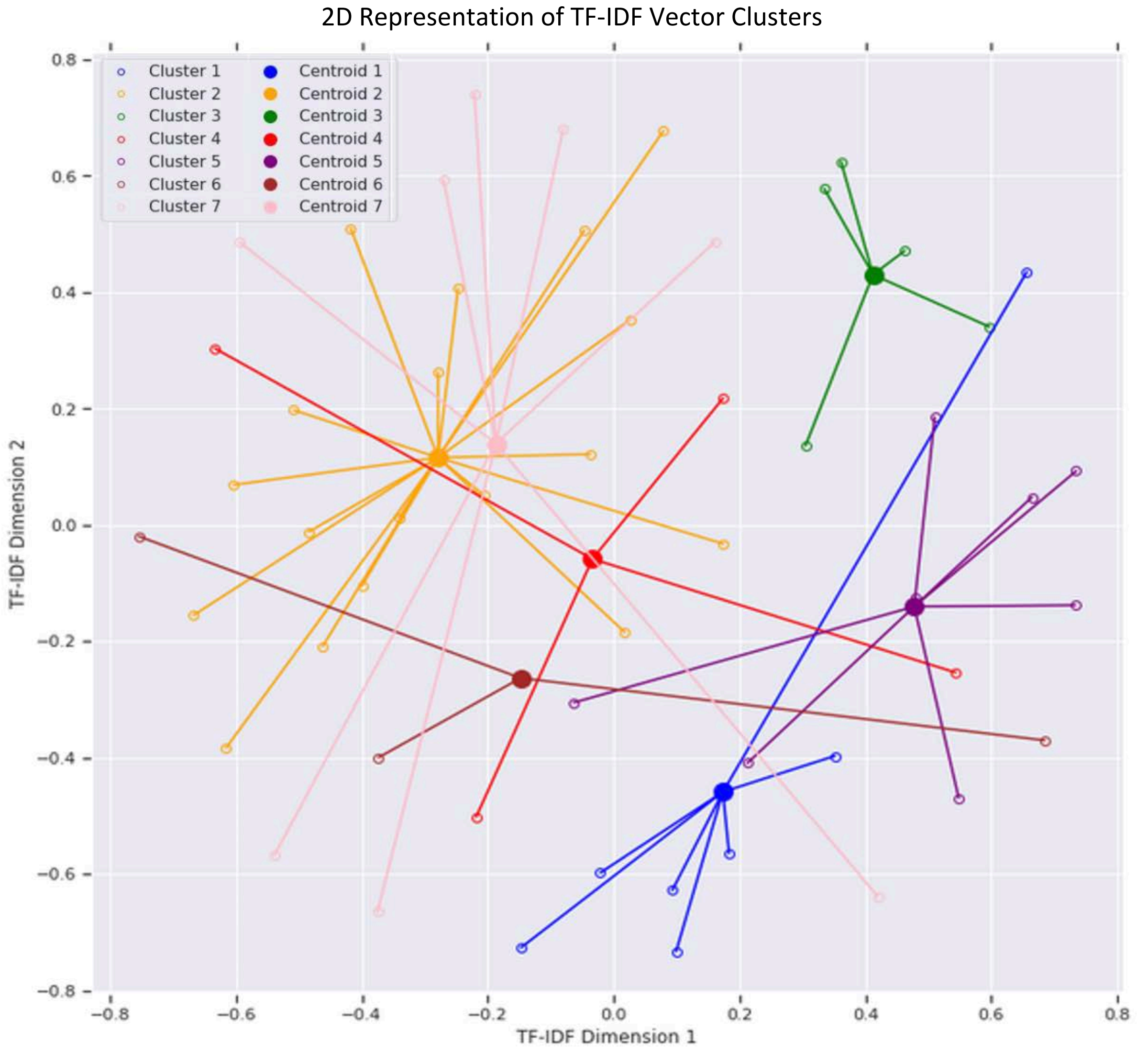


Figure 3. This two-dimensional representation of the TF-IDF vectors for each document in the corpus helps give a sense of how we might eventually want our topic maps to be distributed. While clustering is of course different than topic mapping, this nonetheless gives us a sense that seven is an appropriate number of topics and gives us a sense for the relative topic weightage we might expect.

Topic Mapping

Finally, we performed LDA topic mapping for each of our experiments. As noted in the Literature Review, the output of this model consists of both a term distribution per topic and a topic distribution per document. Both of these proved essential not only in understanding why each model worked the way that it did, but also in the formulation of our proposed framework for making the connection between the LDA input space and model performance and for explaining the sensitivity of the LDA model to specific, seemingly-small changes in term vector preparation.

In the Results section below, we walk through each of our six experiments and show what each particular set of data preparation choices does to the LDA topic mapping model. For each model, we first examined the probability distribution across topics for each document in our corpus. Second, we looked at the word cloud for each topic to see the most common words associated with each topic across the entire corpus. And third, we examined the overall topic weightage distribution across the corpus. These, all reported on next, would become the foundation of our analysis and the basis for how we determined LDA model performance for each experiment.

Results

We ran six experiments, beginning with a baseline model and incrementally adding more sophisticated feature engineering techniques into each successive model run, examining the LDA model output for each one. To evaluate model performance in each experiment, we focused on the heat map showing all of the documents' topic distribution probability vectors, the word clouds showing the most common words associated with each topic, and the bar graph comparing corpus-wide topic weightage for each of the topics.

For the baseline Model 0 run, the data preparation included a few basic steps such as removing punctuation and special characters, removing non-alphabetic characters, and lowercasing all the letters, but notably did not include any of our experimental steps like removing very short words, removing stop words, lemmatizing, creating equivalence classes, or refining the stop words list and equivalence class dictionary. While this model is comically useless as an exercise in generating representative topic maps, it is quite instructive in showing what the LDA algorithm will do with useless input tokens if they are not explicitly accounted for. As seen in the below Figure 4, almost all of the documents' topic probabilities are allocated into a topic characterized by the presence of the terms “the”, “to”, and “of”.

Model 0 Word Cloud and Topic Probability Distribution

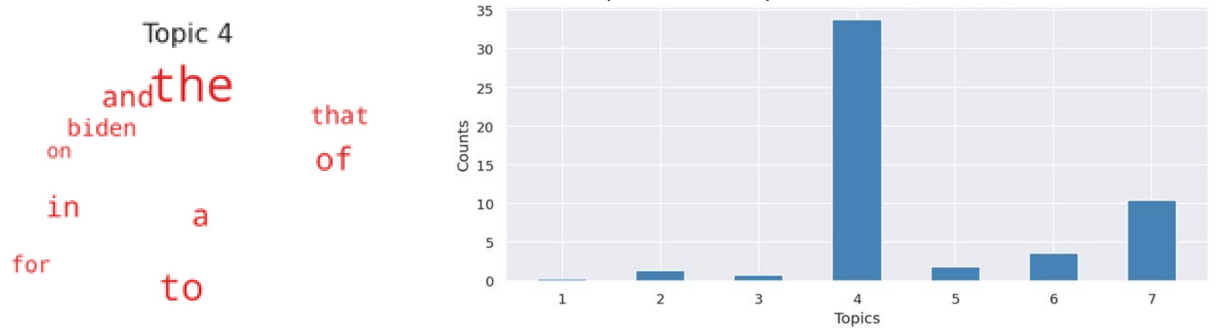


Figure 4. Left panel: Word cloud associated with Topic 4; Right panel: Topic weightage distribution. This figure demonstrates what an LDA topic model will do with empty, meaningless tokens if they are not specifically accounted for and removed (as evidenced by these plots from Model 0).

Experiment 1 removes tokens of fewer than four letters, and Experiment 2 further removes NLTK's default library of stop words, combining to effectively eliminate the above problem, as evidenced by Model 2's word clouds found in Appendix III. But we still see some big problems with Model 2. For example, the corpus token "loans" (plural) is one of the big drivers of Topic 1 allocation, but the token "loan" (singular) leads to Topic 5 allocation. The model is interpreting these words as having different meanings, and as a result, cannot make for proper distinctions between topics. The word clouds demonstrate that two of the seven topics (Topics 1 and 5) are on student loans, and another two (Topics 2 and 7) are focused on climate change. At the same time, the mappings are missing a coherent topic on the COVID fight, which we know from our ontological worldview is one of the most, if not the most, important policy priority of the Biden Administration; the words “covid” and “vaccine” are nowhere to be found in any of the word clouds, for example.

Model 3, after lemmatization is applied, is the first one whose LDA results appear to be based on mostly meaningful tokens. But that does not mean our job was done. If anything, the main result here is that the model is still doing a very poor job of even approximating our ground truth vision of the corpus and the policy priorities of the Biden Administration. We observe a very imbalanced weightage of topic probability mass across the corpus, one that is incompatible with our ontology (see Figure 5 below), and very undefined topics. Model 4, for its part, with its introduction of five equivalence classes that we created, did not significantly change the above.

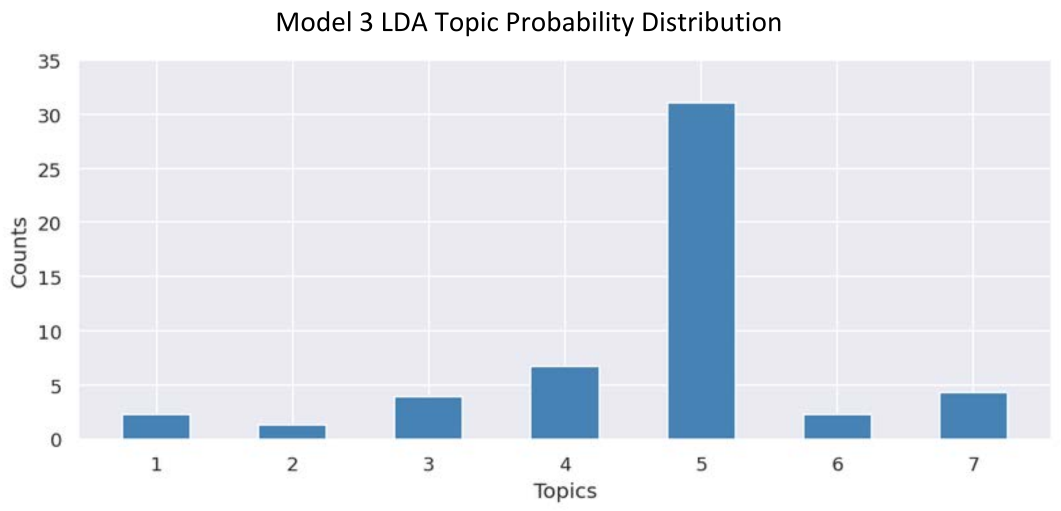
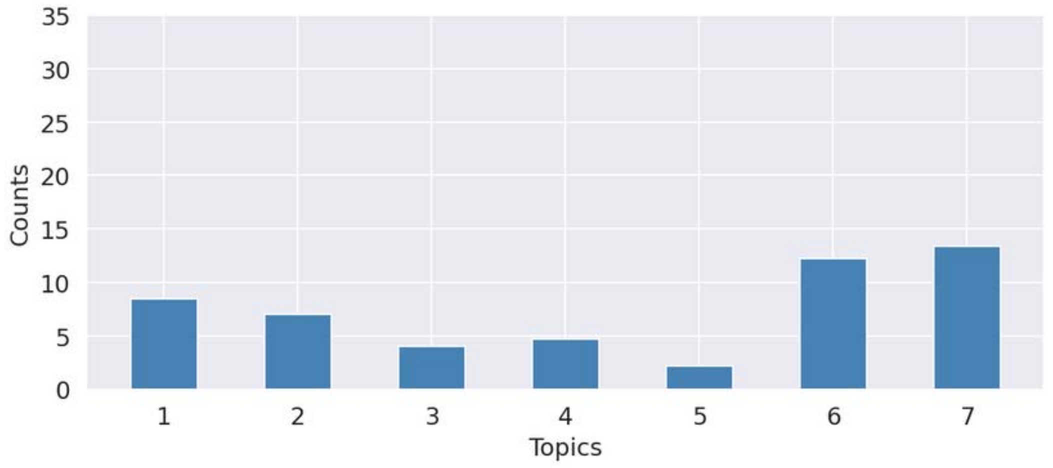


Figure 5. This topic weightage distribution plot from Model 3 shows that, even after several algorithmic preprocessing steps have been applied (short tokens removed, default stop words removed, and lemmatization), the model is still insufficient. Here we see that Model 3 still fails to produce reasonable, distinct topics.

Our real breakthrough came with Model 5. Experiment 5 was developed over dozens of iterations of manually refining both a curated list of twenty-five additional stop words to exclude from our term vectors and also eight additional equivalence classes on top of the five already developed for Experiment 4. Throughout this very labor-intensive process, we were able to develop a topic mapping that began to bear some resemblance to our initial ontology.

This process of equivalence class and stop word list refinement was anything but linear, however. The results were extremely volatile from one iteration of the model to the next. Below we show how the re-inclusion of just a single stop word into the term vectors in Model 5 completely disrupts the delicate equilibrium of Model 5's final probabilistic allocation. The actual Model 5 results are in the top panel in Figure 6 below, and in the bottom panel, we show the same model with just one stop word ("administration") reintroduced into the input space. While the final Model 5 topic mappings are quite reasonable, this incredible volatility and sensitivity to such a small change in the input space is cause for further analysis.

Model 5 LDA Topic Probability Distribution with Administration Stop Word



Model 5 LDA Topic Probability Distribution without Administration Stop Word

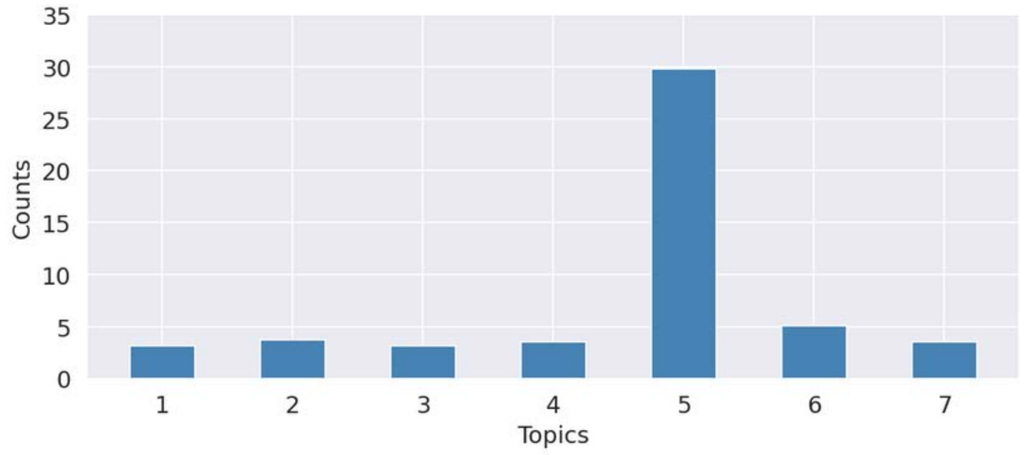
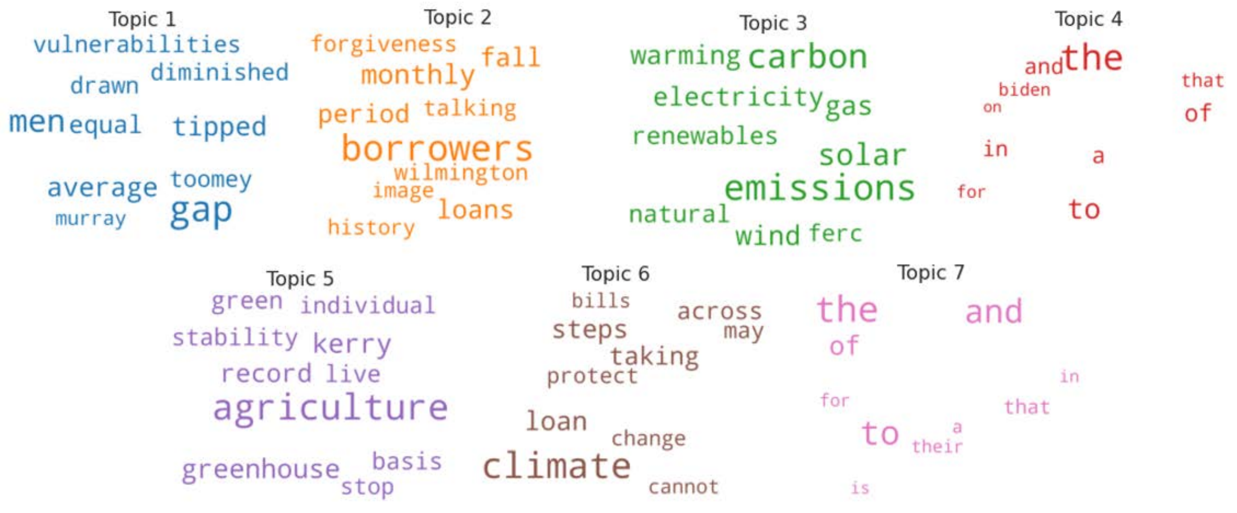


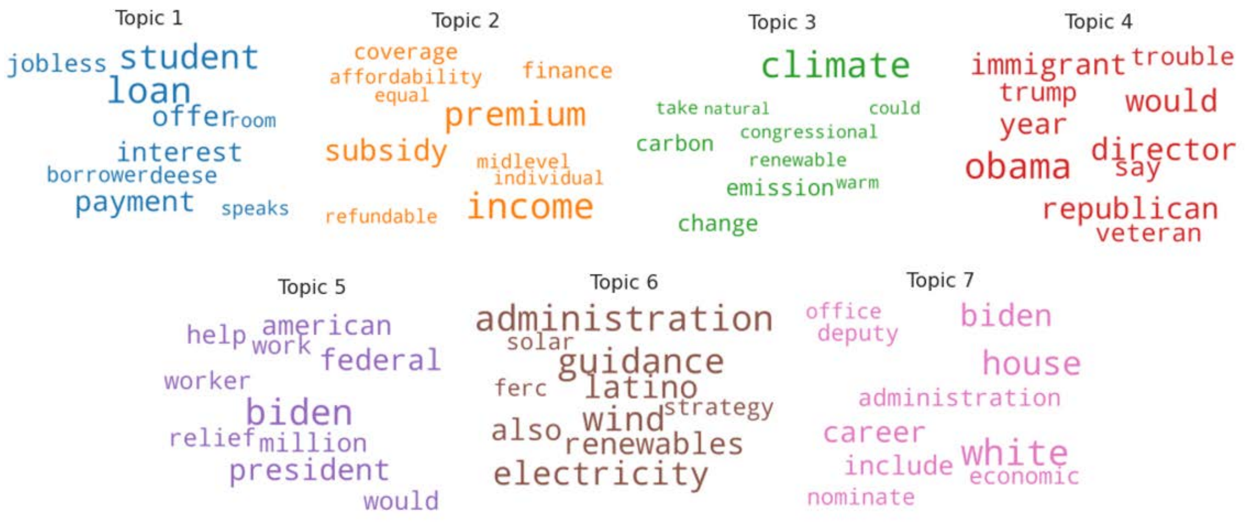
Figure 6. Top panel: "Administration" included as a stop word; Bottom panel: "Administration" not included as a stop word. This comparison of topic weightage distributions from two different iterations of Model 5 development demonstrates the sensitivity of the LDA algorithm to small changes in its input space. The only difference between the two models represented here is the presence or absence of one additional stop word.

The development of model improvement over the course of these experiments can be summarized in the below tables. In Figure 7, we show word clouds from Models 0, 3, and 5. These three models represent the results of our data preparation steps at baseline, after a few algorithmic interventions, and then finally after serious manual interventions. Here we can see the progression from a topic mapping based primarily on meaningless tokens, to one based on actually-important words (mostly) but without any clear topic definition, to one that finally looks a little bit like the policy priorities of the Biden Administration.

Word Cloud of Model 0



Word Cloud of Model 3



Word Cloud of Model 5

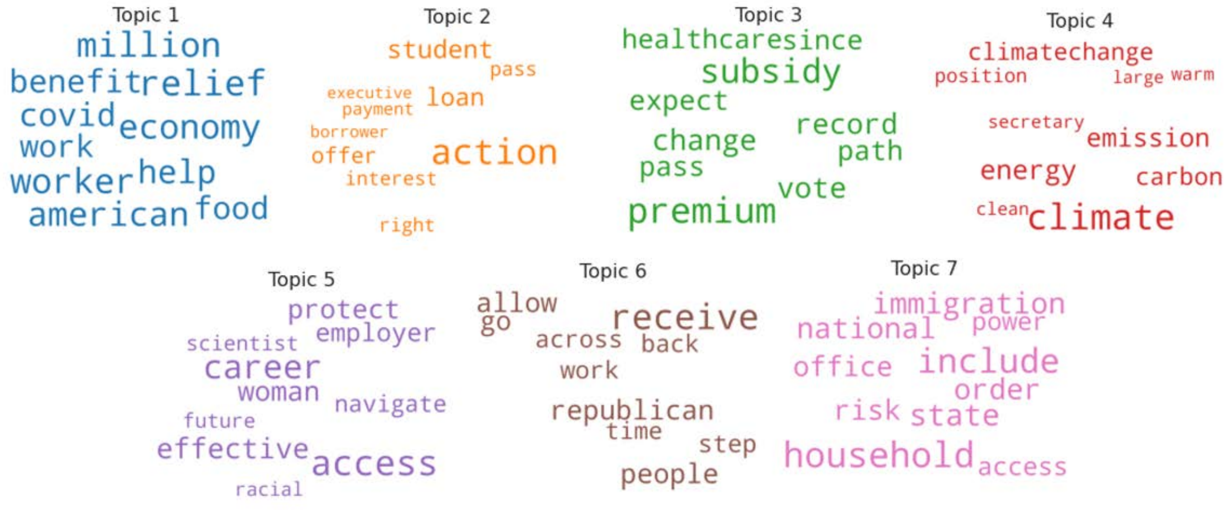


Figure 7. This figure shows the word clouds coming from Models 0, 3, and 5, respectively. It demonstrates the progression of the model from baseline (Model 0), to one that includes a few algorithmic interventions (Model 3), to the final model that also includes serious manual interventions (Model 5).

In Figure 8 below, we also show a comparison of the topic weightage bar graphs for these same three models. Again, we see a similar progression from one model to the next, culminating in a final Model 5 whose topic weightage distribution resembles our ground truth prior.

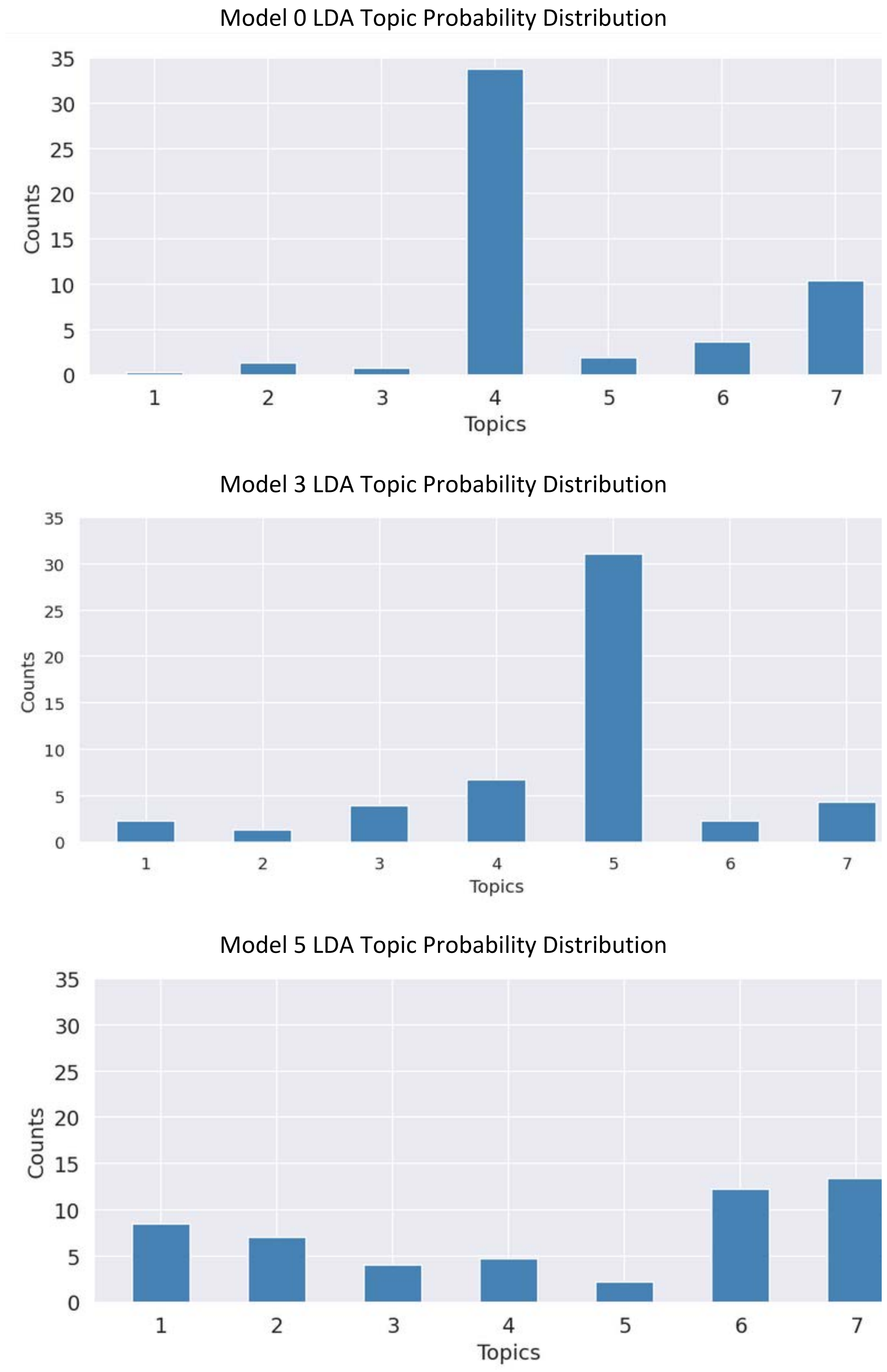


Figure 8. Model 0, 3, and 5 – Like Figure 7 before it, this figure shows the topic weightage distribution graphs coming from Models 0, 3, and 5. Again, it shows the improvement of the model output.

We stopped Experiment 5 development when the topics generated began to resemble our ontology-driven view of ground truth. Not until this point did the LDA output reflect truly coherent, distinct topics that overlapped with our ontology. The below table shows Model 5's seven topics.

Table 2. LDA Topic Descriptions.

Topic Number	*Description*
One	The fight against COVID-19
Two	Student loans and other areas for executive actions
Three	Healthcare policy beyond the pandemic
Four	Climate change
Five	Women's issues and justice issues
Six	Economic stimulus and how to pass the overall agenda
Seven	Immigration policy and trade policy

In Figure 9 we reproduce our initial ontology, color-coding it to show where the topics from Table 2 above map onto it.

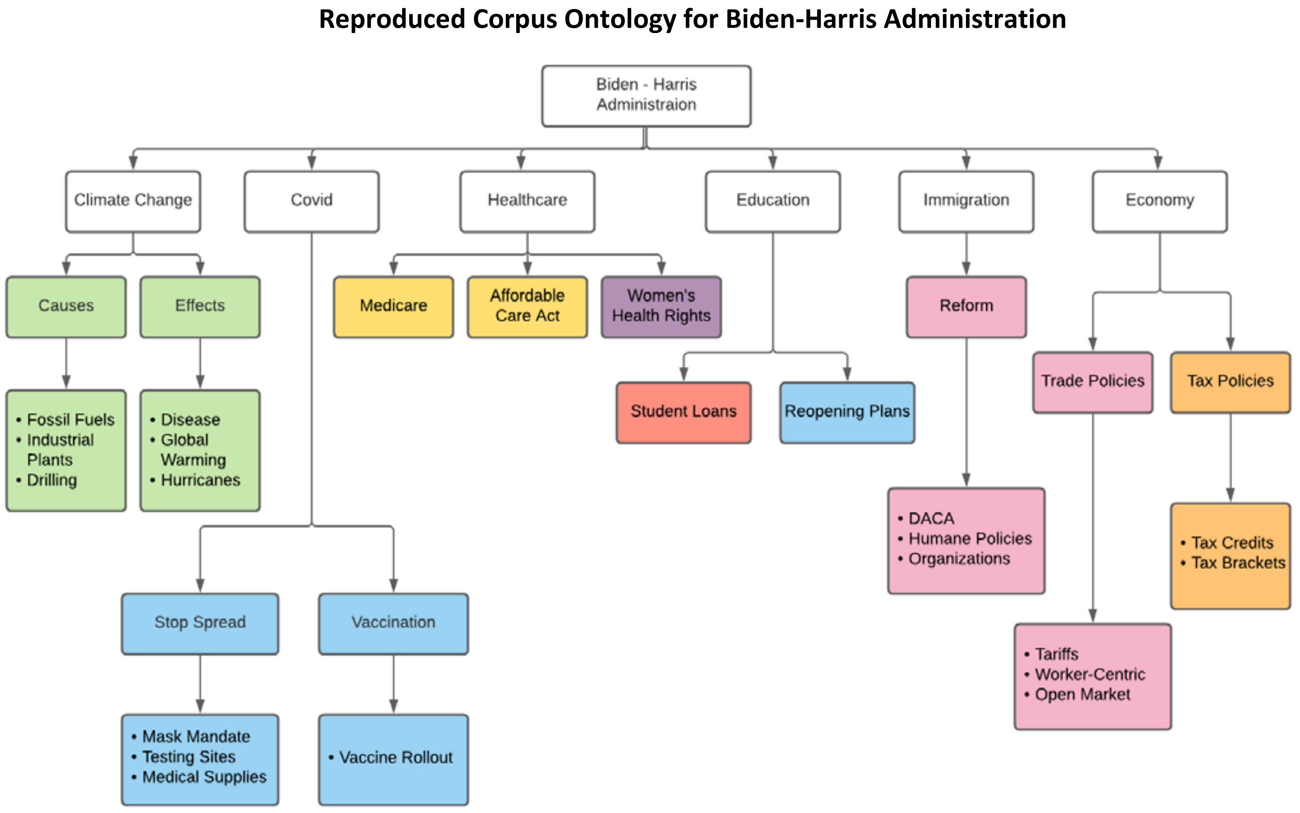

Reproduced Corpus Ontology for Biden-Harris Administration
Biden - Harris Administraion
Climate Change
Covid
Healthcare
Education
Immigration
Economy
Causes
Effects
Medicare
Affordable Care Act
Women's Health Rights
Reform
Trade Policies
Tax Policies
Student Loans
Reopening Plans
• Fossil Fuels • Industrial Plants • Drilling
• Disease • Global Warming • Hurricanes
• DACA • Humane Policies • Organizations
• Tax Credits • Tax Brackets
Stop Spread
Vaccination
• Tariffs • Worker-Centric • Open Market
• Mask Mandate • Testing Sites • Medical Supplies
• Vaccine Rollout

Figure 9. This reproduction of our initial ontology of the Biden-Harris Administration's policy priorities is now color-coded to show where the topics from Table 2 above map onto the ontology.

Interestingly, it took the highly manual feature engineering process of Experiment 5 to begin to see these results. But, as noted above, these results are also quite fragile and quite sensitive to any changes in the input space. Why are the manual data preparation interventions so important to the output? Why is LDA so susceptible to even small changes in the feature space? What does this tell us about the algorithm and how it works? We will tackle these questions in the Analysis and Interpretation section that follows, and in the process propose a framework for understanding the Latent Dirichlet Allocation algorithm.

Analysis and Interpretation

Our discussion begins with this seemingly obvious takeaway: our Latent Dirichlet Allocation algorithmic results were extremely sensitive to the choices made in preparing our reference term vectors. As we walked through our various experiments in the Results section above, the LDA model results varied greatly–improving along the way, but with dramatic fluctuations–as our incremental feature engineering steps made for document vectorizations more strongly suited to topic extraction.

The steps taken to get from Model 0 to Model 2 can be collectively thought of as the low-hanging fruit of NLP data preparation. Even many introductory-level tutorials will algorithmically remove punctuation and special characters, remove very short tokens like "of" and "to", and remove some baseline set of stop words. We included these early models as experiments, though, mostly to present the counterfactual of what would happen if these steps are not applied. The results presented in the previous section showed how the LDA algorithm will grasp on to these meaningless, empty tokens and try to make sense of them when there is no sense to be made, leading to an entirely unworkable model. With the inclusion of these steps plus the lemmatization of Model 3, though, we begin to move towards a mapping of topics that is at least based on meaningful tokens and looks like a reasonable first approximation of our corpus. But still, even at this stage, our model was woefully insufficient.

The biggest improvement we saw actually happened between Model 3 and Model 5. Through Model 3, the feature engineering steps were largely algorithmic. But from Model 3 through 5, the changes in vector representation were entirely based on manual decisions we made. Over dozens of iterations of creating and modifying equivalence classes and adding a custom, curated list of stop words to our earlier NLTK default stop words library, we improved our document vector representations so much so that Model 5 starts to resemble our ground truth and ontology-based representation of our corpus. But it took this incredibly time- and manual labor-intensive process to get to this point. One key finding of our research was just how essential this manual process is for topic modeling.

But why is this manual feature engineering step so important, especially for LDA? Our research offers one suggestion. In particular, the biggest gains in model performance were realized by excluding certain words altogether from the vector of reference terms. Earlier, we touched on TF-IDF and spoke to how it penalizes terms that are too common throughout the corpus because that commonality makes their appearance in certain documents less significant. The LDA model, though, does not use TF-IDF vectors as its input; it instead uses a bag-of-words model for developing its input corpus. Thus, unlike TF-IDF-based models, LDA is susceptible to influence from common yet relatively empty tokens. While this hurts the model's ability to base its probabilistic allocations on the corpus's more information-rich terms, it also presents the modeler with a real opportunity to improve performance by taking the time to manually remove them, effectively applying a document frequency penalty, albeit an extreme one, to a bag-of-words-based topic model. Comparing the results of Model 3 to Model 5 above demonstrates the impact of this step, as only when we remove the influential empty tokens such as "says" and "administration" from our LDA analysis can our model find a more appropriate mapping.

Taking this idea one step further, one of the most important findings of our research is that within the world of clustering and topic mapping, LDA is almost uniquely susceptible and sensitive to changes in the input space because of the probabilistic nature of the algorithm. We propose a framework for understanding LDA that considers the tokens themselves almost as conduits in an interconnected web. Because the algorithm returns both a term distribution per topic and a topic distribution per document, each term does not simply point to a single topic, but rather probabilistically points to multiple, forming a bridge between topics. And because the topics are also not unique within documents, they in turn form bridges between the documents. Thus, when we change even one single input term, such as by including it in an equivalence class or by removing it altogether, it has downstream ripple effects on the other terms it is linked to through the topic distributions and the documents those topics were linked to, as all of the probabilities are now forced to redistribute and reallocate themselves.

Figure 10 on the following page, a network graph of the seven topics and the dominant terms in their term distributions, demonstrates this interconnected nature of our final LDA model. In doing so, it also helps to illustrate why each of the specific decisions made in input vector preparation is so important to the Latent Dirichlet Allocation model in particular, as this networked effect makes it so that a change to any word necessarily impacts so many others, as well.

This probabilistic interconnectedness has its advantages, too. While the above network graph speaks to LDA's term distribution per topic output, it is the model's topic distribution per document return that can inform our interpretation of the relationships between topics. Appendix VI shows a heat map of the probability distribution of topics within each document for our final Model 5, and it is quite useful in demonstrating how certain documents span multiple topics. Documents 39, 46, and 47, for example, are all nominally centered around the idea of economic stimulus, according to our manually-determined ground truth, but it is instructive to see what other topics these documents' probability vectors give weight to. While Topic 6 is the one most closely and directly related to the idea of stimulus, the probability vectors for these documents also show high probability masses within Topic 1. Topic 1 focuses on the pandemic, and while the fight against COVID-19 has a strong healthcare focus, the apparent linkage between Topics 6 and 1 in these documents is the model's way of demonstrating that getting past the pandemic and reviving our economy are inexorably linked. We can look to individual terms in our model like "relief" and see how the term distributions for both Topics 1 and 6 strongly favor this same term. We see this same relationship between COVID relief and economic relief in our

Topic-Word Network Map

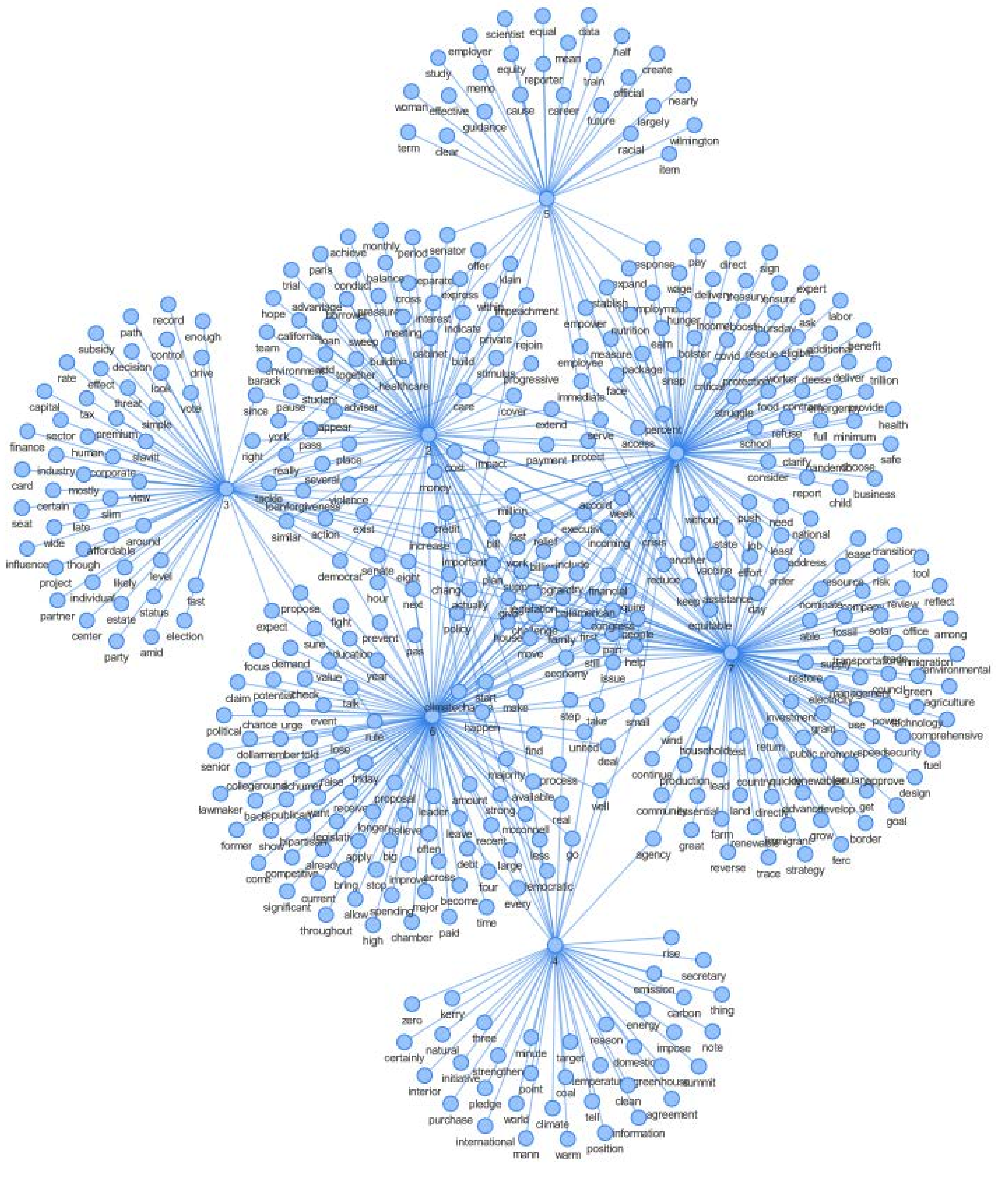


Figure 10. This topic-word network map is meant to be viewed holistically so as to demonstrate the interconnected web that a topic map represents, where all the terms and topics are linked to one another.

ontology, as "Stimulus Proposals" falls directly under "COVID-19 Fight"; this worldview confirmation of an important model result contributes to our confidence in Model 5. At the same time, this example is also illustrative of one of the biggest advantages of topic mapping over clustering, that it can discover and reveal to us such connections between topics and ideas in our corpus.

Just as individual terms in our input space serve as the bridges between topics, so too do they serve as the link between documents. We gave the example in the previous paragraph of how the token "relief" is an essential part of linking Topics 1 and 6, but it is similarly instrumental in making the connections between certain documents. Of course the term is useful in connecting the documents about COVID-19 and economic relief, but it also helpful in making less obvious connections, like to the six documents in our corpus about Biden's First 100 Days, which could conceivably be about any of our seven topics. In doing so, it shows us that this idea of COVID/economic relief is the top priority of the Administration. Again, this all goes to say: in the interconnected web of topics and documents, it is the input tokens themselves that are the glue. This only further goes to show why term vector representation is so vital in LDA, and further goes to draw a straight line from those manual input space engineering decisions to the end-result topic mapping.

But the essential nature of term vector representation in LDA also highlights one of the major limits of our research. When the corpus documents themselves do not contain the specific words needed to detect certain topics or make the aforementioned connections, the model has zero basis for extrapolation. The model only has the vocabulary it has, and it relies on it exclusively for being that connective tissue and ultimately for coming up with the final topic maps. That makes topic mapping simultaneously over-sensitive to both the specific vocabulary choices within our corpus documents themselves and also the particularities of the decisions we make in preparing them. In the Directions for Future Work section below, we postulate a few remedies for this fundamental issue.

Finally, our last takeaway focuses on the immense power of the modeler in LDA topic mapping. We have demonstrated the connection between term vector representation and model effectiveness, and also shown just how important the manual decisions in particular are in producing worthwhile reference term vectors. This puts the modeler in a difficult position, as there is no theoretical end to the number of tweaks and iterations one can make and test, each of which can have a very real impact. This is simultaneously exhilarating and terrifying. We began this discussion with the dizzying observation of just how absolutely sensitive this model is to the smallest choices the user can make. And so we end humbled by the importance of the manual feature engineering process, but now also with a much stronger sense of why Latent Dirichlet Allocation is so volatile with respect to its connective tissue.

Conclusions

In this analysis, we set out to make clear the direct connection between the choices made in reference term vector preparation and Latent Dirichlet Allocation model performance. And while feature engineering is always vital in Natural Language Processing applications, we have demonstrated its unique importance in LDA topic mapping.

We did so by creating an incremental series of experiments in which we layered one data preparation step after another, culminating in the final manual step of iteratively customizing our equivalence classes and an expanded list of additional stop words to exclude. We saw how the model seemed to work better with each successive experiment. But it was not until after the final manual step when the topic mapping output began to resemble a reasonable facsimile of the ontologically-based ground truth conception of our corpus.

Our research delved into the question of why the manual feature engineering steps are so significant for LDA, and we tied it back to the differences between bag-of-words and TF-IDF document vectorization, with the former lacking a penalty mechanism for common yet insignificant words. But the primary reason the LDA model is so susceptible to changes in the input space has to do with the probabilistic nature of its development, with the terms that make up the corpus serving as the links between the topics and between the documents. When that connective tissue is manipulated, the model has to redistribute and reallocate all of its probabilities, and the ripple effects of that process can be significant. This makes the LDA algorithm extremely sensitive to even very small changes in its reference term vector representations.

Latent Dirichlet Allocation topic modeling is a powerful statistical approach for organizing and understanding the relationships between topics and documents in a corpus. Our research advanced the understanding of what truly drives this model and why the connection between data preparation and model performance is both so strong and so volatile.

Directions for Future Work

In this work, we sought to understand why the LDA model is so uniquely sensitive to changes in its input space. In our future work, we seek to help solve some of this instability and volatility.

One primary source of volatility in this research was the small size of our corpus. Our corpus consisted of only 53 documents, and those documents did not exactly resemble a representative sample of the topic space. As part of our future work on this topic, we will want to build a considerably larger corpus, one that samples more evenly from all of the biggest priorities of the Biden Administration. Improving our sample size will alone make for less volatile models, as each term will begin to approach a more characteristic representation.

But the biggest direction of future work is to investigate utilizing the lda2vec model (Moody 2016). This model is an expansion of the word2vec model that simultaneously learns word, document, and topic vectors, effectively building a topic map on top of word embeddings. Our interest in this model stems from our desire to see if the use of word embeddings as inputs can reduce the volatility in our topic mappings.

Finally, if we can build a sufficiently large corpus, we wish to experiment with recurrent neural networks (RNNs) to group or cluster our corpus documents. With their ability to capture sequential data and have “memory”, RNNs have advantages over approaches for clustering that do not take word order into account, like the ones based on bag-of-words or TF-IDF vectors.

Author Information

Corresponding Authors

MH, marcus.hilliard@gmail.com, marcushilliard2021@u.northwestern.edu; RH, Rikita1994@gmail.com, rikitahirpara2021@u.northwestern.edu; DM, doron.meizlik@gmail.com, doronmeizlik2021@u.northwestern.edu; KZ, zink.kath@gmail.com, kathrynzink2023@u.northwestern.edu

Author Contributions

The manuscript was written through equal contributions of all authors. All authors have approved the final version of the manuscript.

Acknowledgment

This article is the result of course work produced during the Master's course in Natural Language Processing at Northwestern University.

Conflict of Interest Statement

The authors report no conflict of interest.

Abbreviations

MH, Marcus Hilliard; RH, Rikita Hirpara; DM, Doron Meizlik; KZ, Katie Zink.

References

Ayo, Femi Emmanuel, Olusegun Folorunso, Friday Thomas Ibharalu, Idowu Ademola Osinuga, and Adebayo Abayomi-Alli. “A Probabilistic Clustering Model for Hate Speech Classification in twitter.” Expert Systems with Applications, 2021, 114762. https://doi.org/10.1016/j.eswa.2021.114762.

Baharudin, Baharum, Lam Hong Lee, and Khairullah Khan. “A Review of Machine Learning Algorithms for Text-Documents Classification.” Journal of Advances in Information Technology 1, no. 1 (2010). https://doi.org/10.4304/jait.1.1.4-20.

Bíró, István, Jácint Szabó, and András Benczúr. “Latent Dirichlet Allocation in Web Spam Filtering.” In Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web, 29–32. ACM, 2008. https://doi.org/10.1145/1451983.1451991.

Bojanowski, Piotr, Edouard Grave, Armand Joulio, and Tomas Mikolov. 2016. "Enriching Word Vector with Subword Information." Transactions of the Association for Computational Linguistics 5, no. 1 (July 2016): 135-146. https://arxiv.org/abs/1607.04606.

Brownlee, J. “A Gentle Introduction to the Bag-of-Words Model.” Machine Learning Mastery. October 9, 2017. Accessed February 24, 2021. https://machinelearningmastery.com/gentle-introduction-bag-words-model/ .

Carlberger, Johan, Hercules Dalianis, Martin Hassel, and Ola Knutsson. “Improving Precision in Information Retrieval for Swedish using Stemming.” In Proceedings of NODALIDA 2001. https://www.aclweb.org/anthology/W01-1703.pdf.

"Classify Text Using SpaCy." Dataquest. April 21, 2020. Accessed March 10, 2021. https://www.dataquest.io/blog/tutorial-text-classification-in-python-using-spacy/.

Filliat, D. “A Visual Bag of Words Method for Interactive Qualitative Localization and Mapping.” In Proceedings 2007 IEEE International Conference on Robotics and Automation, 3921–26. IEEE, 2007. https://doi.org/10.1109/ROBOT.2007.364080.

Hamilton, Leah and Jacob Lahne. “Fast and Automated Sensory Analysis: Using Natural Language Processing for Descriptive Lexicon Development.” Food Quality and Preference 83 (2020): 103926. https://doi.org/10.1016/j.foodqual.2020.103926.

Iyyer, Mohit, Peter Enns, Jorden Boyd-Graber, and Philip Resnik. 2014. "Political ideology detection using recursive neural networks." 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, MD, USA. 1113-1122.

Kim, Donghwa, Deokseong Seo, Suhyoun Cho, and Pilsung Kang. “Multi-Co-Training for Document Classification Using Various Document Representations: TF–IDF, LDA, and Doc2Vec.” Information Sciences 477 (2019): 15–29. https://doi.org/10.1016/j.ins.2018.10.006.

Lau, Jey Han, and Timothy Baldwin. “An Empirical Evaluation of Doc2vec with Practical Insights into Document Embedding Generation,” 2016.

Le, Quoc V, and Tomas Mikolov. “Distributed Representations of Sentences and Documents,” 2014.

Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. “The Global k-Means Clustering Algorithm.” Pattern Recognition 36, no. 2 (2003): 451–61. https://doi.org/10.1016/S0031-3203(02)00060-2.

Manning, Christopher, Prabhakar Raghavan, and Hinrich Schütze. “Introduction to Information Retrieval.” New York: Cambridge University Press, 2008.

Mikolov, Thomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey. Dean. 2013. "Distributed Representation of Words and Phrases and Their Compositionality." Edited by L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger C.J.C. Burges. Advances in Neural Information Processing Systems 26 (NIPS 2013) (Red Hook, NY: Curran Associates) 2: 3111-3119.

Moody, Christopher E. “Mixing Dirichlet Topic Models and Word Embeddings to Make Lda2vec,” 2016.

Naumov, Stanislav, Grigory Yaroslavtsev, and Dmitrii Avdiukhin. “Objective-Based Hierarchical Clustering of Deep Embedding Vectors,” 2020.

Paul, Snigdha, Mini Tandon, Nisheeth Joshi and Iti Mathur. “Design of a Rule Based Hindi Lemmatizer.” 2013.

Suppiah, Haresh, Richard Swinbourne, Jericho Wee, Vanes Tay, and Paul Gastin. “Sleep Characteristics of Elite Youth Athletes: A Clustering Approach to Optimize Sleep Support Strategies.” International Journal of Sports Physiology and Performance. 2021 Feb:1-9. DOI: 10.1123/ijspp.2020-0675.

Wang, Huizhen. "Introduction to Word2vec and its application to find predominant word senses." Nanyang Technological University. Accessed February 2021. http://compling.hss.ntu.edu.sg/courses/hg 7017/pdf/word2vec%20and%20its%20application%20to%20wsd.pdf.

Wang, Di, Marcus Thint, and Ahmad Al-Rubaie. “Semi-Supervised Latent Dirichlet Allocation and Its Application for Document Classification.” In Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, 3:306–10. IEEE Computer Society, 2012. https://doi.org/10.1109/WI-IAT.2012.211.

Xue, Bai, Chen Fu, and Zhan Shaobin. 2014. "A study on sentiment computing and classification of sina weibo with Word2vec." 2014 IEEE International Congress on Big Data. Anchorage, AK, USA. 358-363. doi:10.1109/BigData.Congress.2014.59.

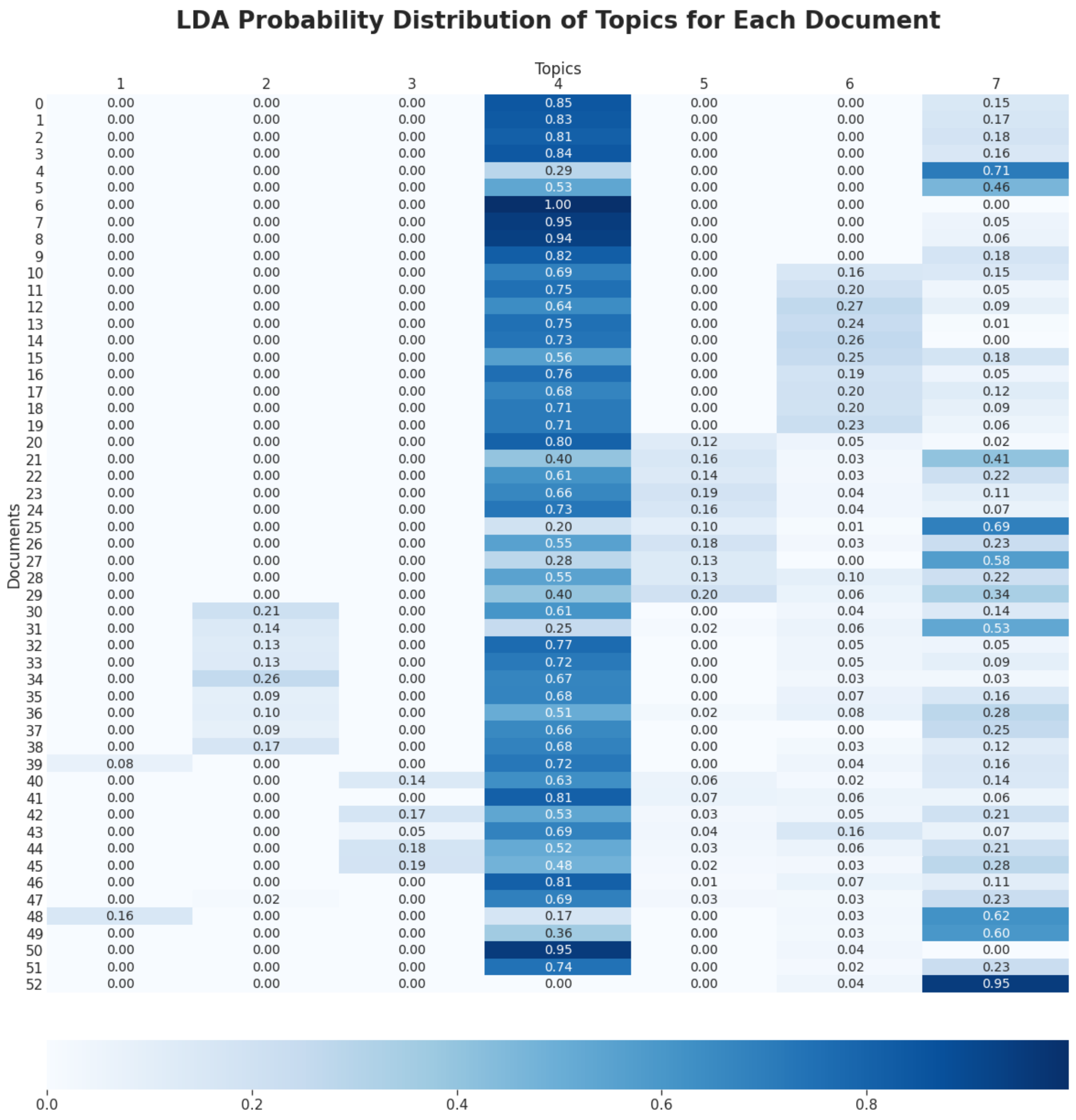
Yildirim, S. “Hierarchical Clustering — Explained.” Towards Data Science. April 3, 2020. Accessed March 1, 2021. https://towardsdatascience.com/hierarchical-clustering-explained-e58d2f936323.

Zhang, Yun-tao, Gong Ling, Yong-cheng Wang. “An Improved TF-IDF Approach for Text Classification.” Journal of Zhejiang University. A. Science 6, no. 1 (2005): 49–55. https://doi.org/10.1007/BF02842477.

Appendix I

Model Zero Results

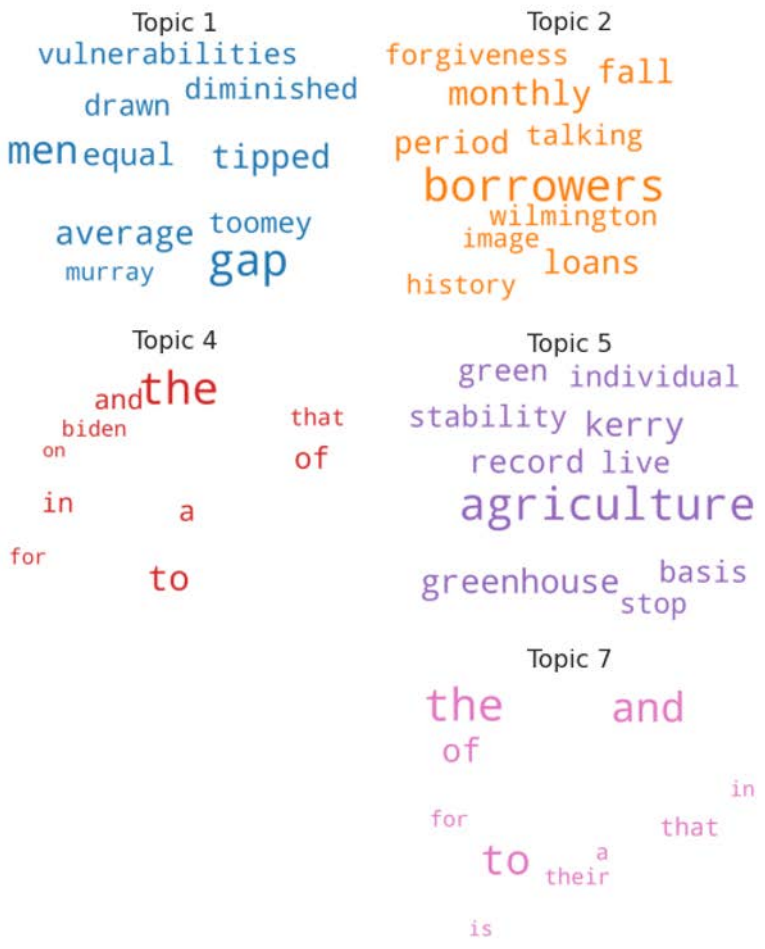
Heat Map of Model 0



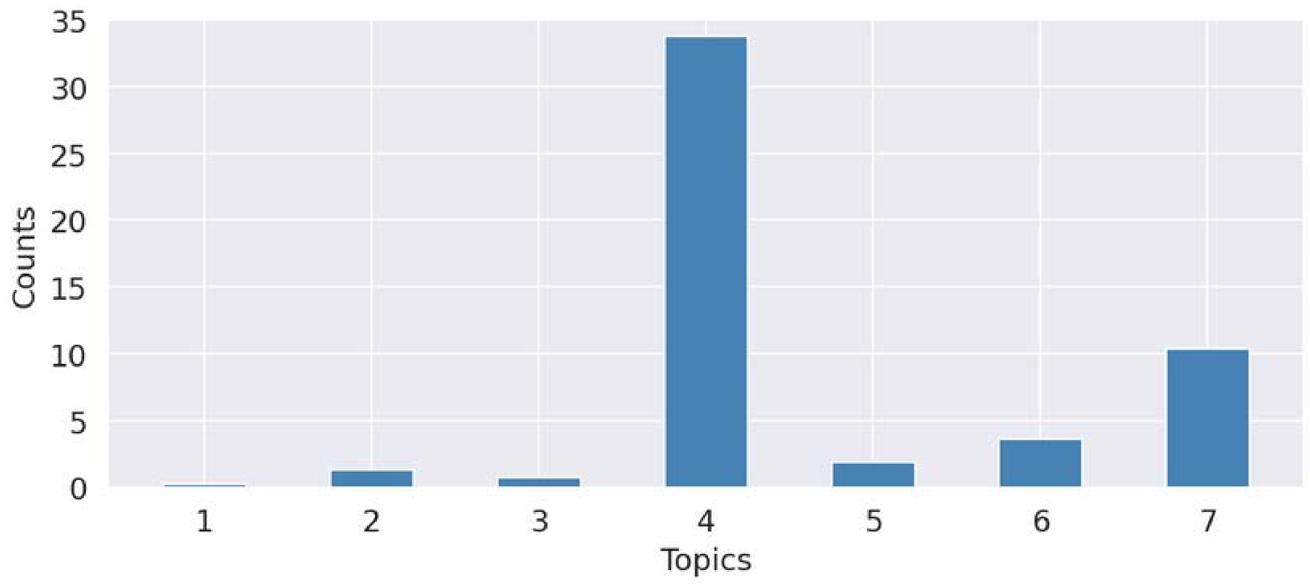
LDA Probability Distribution of Topics for Each Document

Documents	1	2	3	4	5	6	7
0	0.00	0.00	0.00	0.85	0.00	0.00	0.15
1	0.00	0.00	0.00	0.83	0.00	0.00	0.17
2	0.00	0.00	0.00	0.81	0.00	0.00	0.18
3	0.00	0.00	0.00	0.84	0.00	0.00	0.16
4	0.00	0.00	0.00	0.29	0.00	0.00	0.71
5	0.00	0.00	0.00	0.53	0.00	0.00	0.46
6	0.00	0.00	0.00	1.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.95	0.00	0.00	0.05
8	0.00	0.00	0.00	0.94	0.00	0.00	0.06
9	0.00	0.00	0.00	0.82	0.00	0.00	0.18
10	0.00	0.00	0.00	0.69	0.00	0.16	0.15
11	0.00	0.00	0.00	0.75	0.00	0.20	0.05
12	0.00	0.00	0.00	0.64	0.00	0.27	0.09
13	0.00	0.00	0.00	0.75	0.00	0.24	0.01
14	0.00	0.00	0.00	0.73	0.00	0.26	0.00
15	0.00	0.00	0.00	0.56	0.00	0.25	0.18
16	0.00	0.00	0.00	0.76	0.00	0.19	0.05
17	0.00	0.00	0.00	0.68	0.00	0.20	0.12
18	0.00	0.00	0.00	0.71	0.00	0.20	0.09
19	0.00	0.00	0.00	0.71	0.00	0.23	0.06
20	0.00	0.00	0.00	0.80	0.12	0.05	0.02
21	0.00	0.00	0.00	0.40	0.16	0.03	0.41
22	0.00	0.00	0.00	0.61	0.14	0.03	0.22
23	0.00	0.00	0.00	0.66	0.19	0.04	0.11
24	0.00	0.00	0.00	0.73	0.16	0.04	0.07
25	0.00	0.00	0.00	0.20	0.10	0.01	0.69
26	0.00	0.00	0.00	0.55	0.18	0.03	0.23
27	0.00	0.00	0.00	0.28	0.13	0.00	0.58
28	0.00	0.00	0.00	0.55	0.13	0.10	0.22
29	0.00	0.00	0.00	0.40	0.20	0.06	0.34
30	0.00	0.21	0.00	0.61	0.00	0.04	0.14
31	0.00	0.14	0.00	0.25	0.02	0.06	0.53
32	0.00	0.13	0.00	0.77	0.00	0.05	0.05
33	0.00	0.13	0.00	0.72	0.00	0.05	0.09
34	0.00	0.26	0.00	0.67	0.00	0.03	0.03
35	0.00	0.09	0.00	0.68	0.00	0.07	0.16
36	0.00	0.10	0.00	0.51	0.02	0.08	0.28
37	0.00	0.09	0.00	0.66	0.00	0.00	0.25
38	0.00	0.17	0.00	0.68	0.00	0.03	0.12
39	0.08	0.00	0.00	0.72	0.00	0.04	0.16
40	0.00	0.00	0.14	0.63	0.06	0.02	0.14
41	0.00	0.00	0.00	0.81	0.07	0.06	0.06
42	0.00	0.00	0.17	0.53	0.03	0.05	0.21
43	0.00	0.00	0.05	0.69	0.04	0.16	0.07
44	0.00	0.00	0.18	0.52	0.03	0.06	0.21
45	0.00	0.00	0.19	0.48	0.02	0.03	0.28
46	0.00	0.00	0.00	0.81	0.01	0.07	0.11
47	0.00	0.02	0.00	0.69	0.03	0.03	0.23
48	0.16	0.00	0.00	0.17	0.00	0.03	0.62
49	0.00	0.00	0.00	0.36	0.00	0.03	0.60
50	0.00	0.00	0.00	0.95	0.00	0.04	0.00
51	0.00	0.00	0.00	0.74	0.00	0.02	0.23
52	0.00	0.00	0.00	0.00	0.00	0.04	0.95

Word Cloud of Model 0



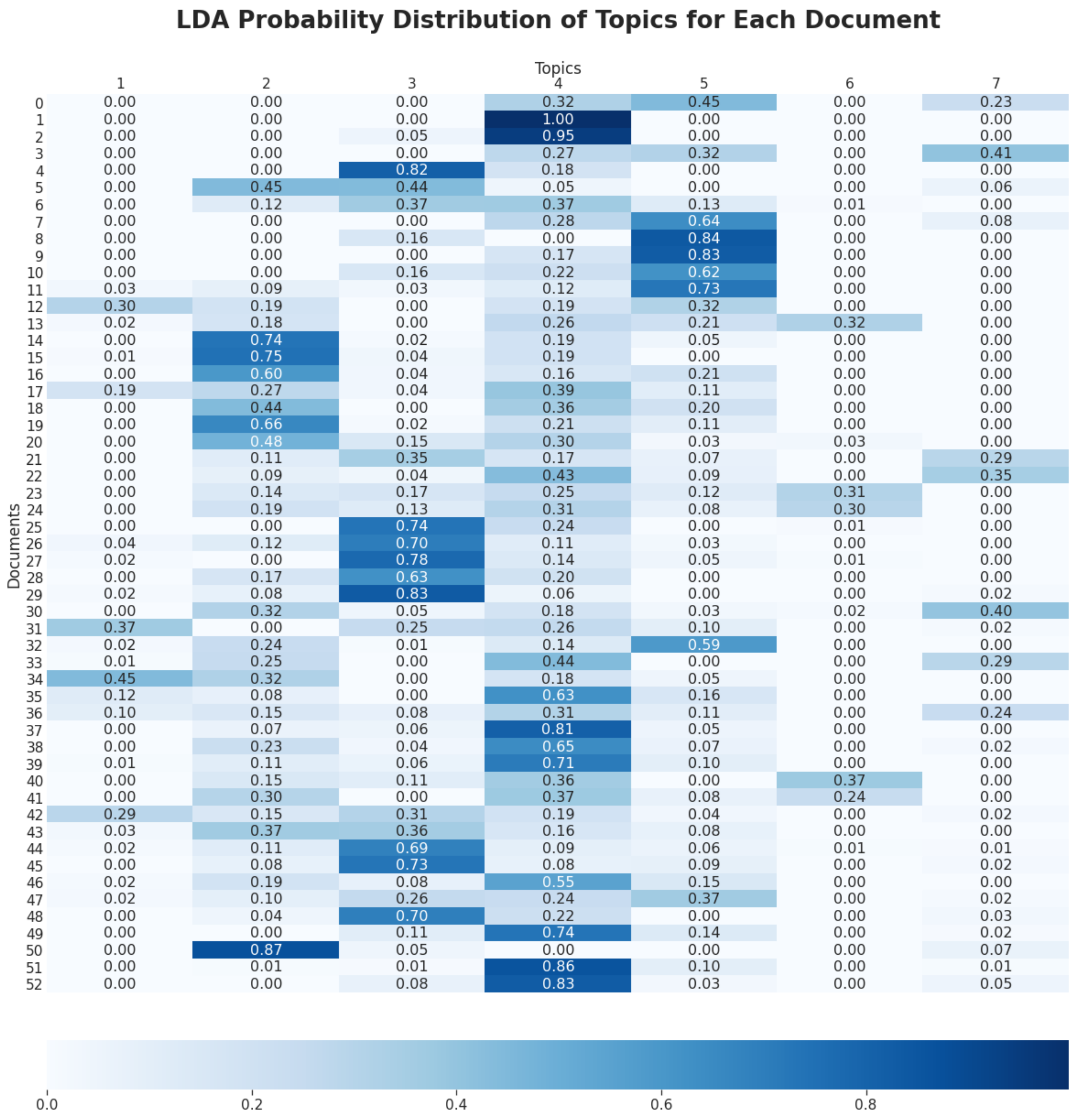
Bar Graph of Model 0



Appendix II

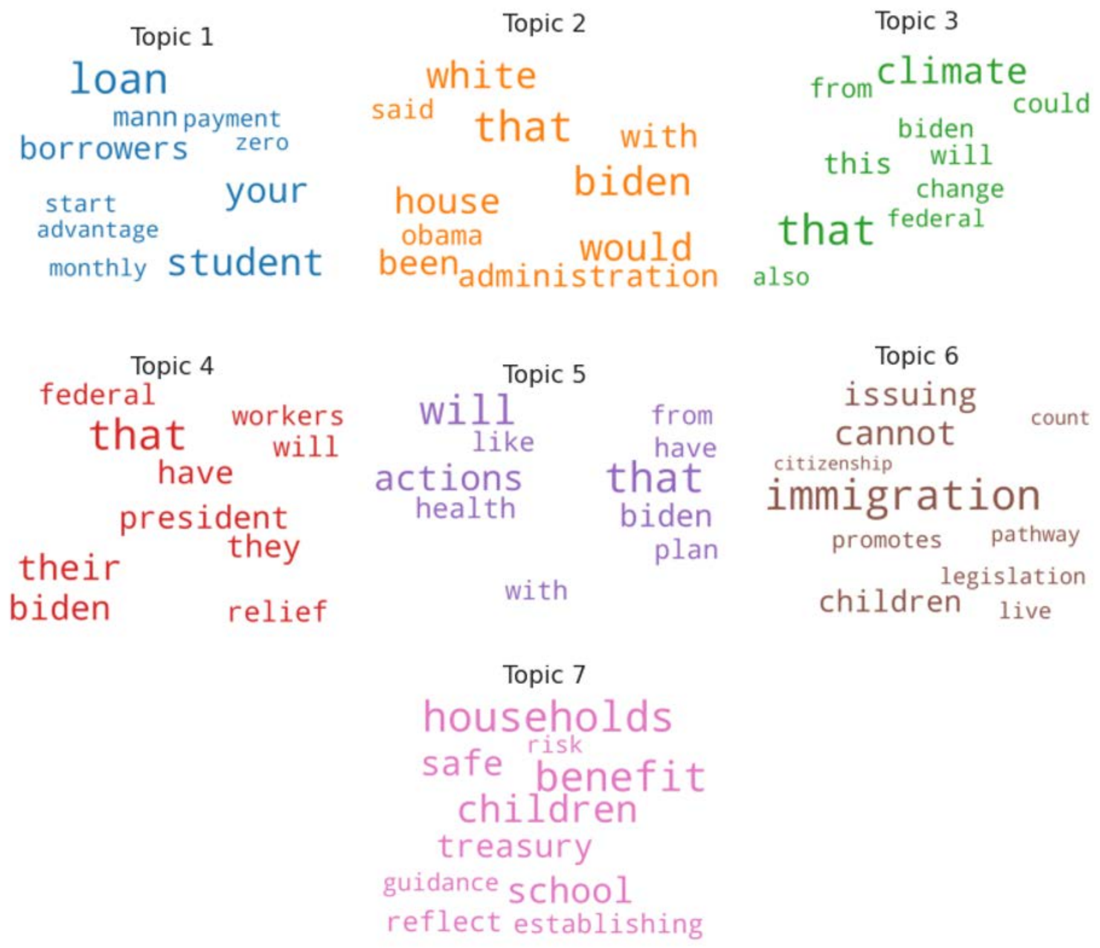
Model One Results

Heat Map of Model 1

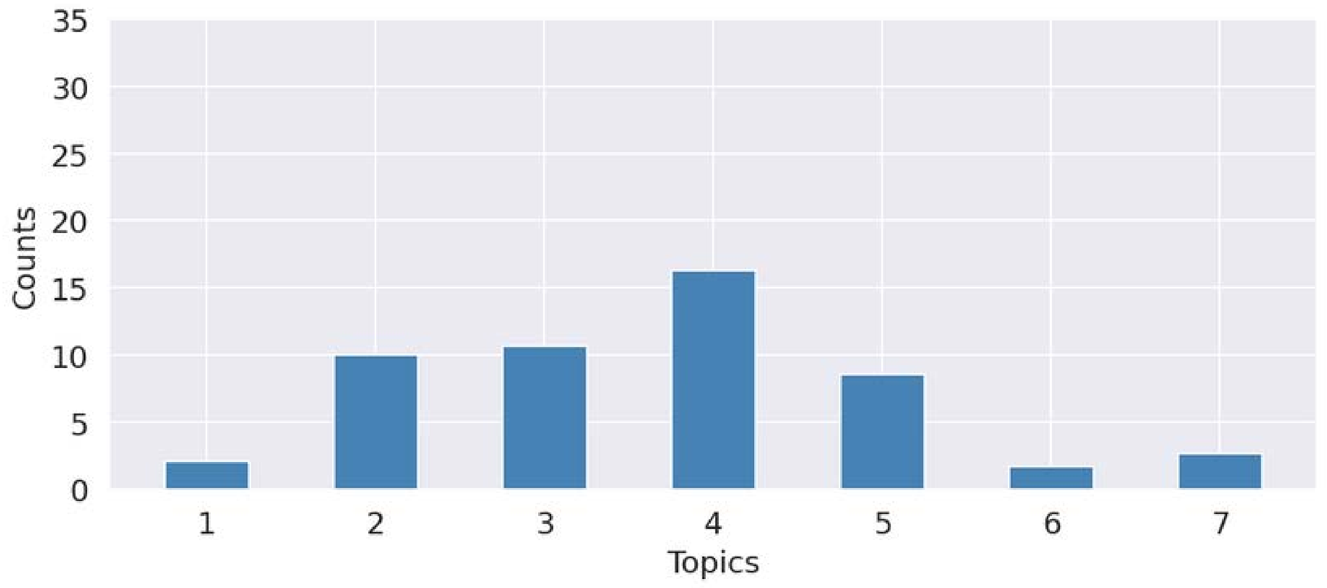


Documents	1	2	3	4	5	6	7
0	0.00	0.00	0.00	0.32	0.45	0.00	0.23
1	0.00	0.00	0.00	1.00	0.00	0.00	0.00
2	0.00	0.00	0.05	0.95	0.00	0.00	0.00
3	0.00	0.00	0.00	0.27	0.32	0.00	0.41
4	0.00	0.00	0.82	0.18	0.00	0.00	0.00
5	0.00	0.45	0.44	0.05	0.00	0.00	0.06
6	0.00	0.12	0.37	0.37	0.13	0.01	0.00
7	0.00	0.00	0.00	0.28	0.64	0.00	0.08
8	0.00	0.00	0.16	0.00	0.84	0.00	0.00
9	0.00	0.00	0.00	0.17	0.83	0.00	0.00
10	0.00	0.00	0.16	0.22	0.62	0.00	0.00
11	0.03	0.09	0.03	0.12	0.73	0.00	0.00
12	0.30	0.19	0.00	0.19	0.32	0.00	0.00
13	0.02	0.18	0.00	0.26	0.21	0.32	0.00
14	0.00	0.74	0.02	0.19	0.05	0.00	0.00
15	0.01	0.75	0.04	0.19	0.00	0.00	0.00
16	0.00	0.60	0.04	0.16	0.21	0.00	0.00
17	0.19	0.27	0.04	0.39	0.11	0.00	0.00
18	0.00	0.44	0.00	0.36	0.20	0.00	0.00
19	0.00	0.66	0.02	0.21	0.11	0.00	0.00
20	0.00	0.48	0.15	0.30	0.03	0.03	0.00
21	0.00	0.11	0.35	0.17	0.07	0.00	0.29
22	0.00	0.09	0.04	0.43	0.09	0.00	0.35
23	0.00	0.14	0.17	0.25	0.12	0.31	0.00
24	0.00	0.19	0.13	0.31	0.08	0.30	0.00
25	0.00	0.00	0.74	0.24	0.00	0.01	0.00
26	0.04	0.12	0.70	0.11	0.03	0.00	0.00
27	0.02	0.00	0.78	0.14	0.05	0.01	0.00
28	0.00	0.17	0.63	0.20	0.00	0.00	0.00
29	0.02	0.08	0.83	0.06	0.00	0.00	0.02
30	0.00	0.32	0.05	0.18	0.03	0.02	0.40
31	0.37	0.00	0.25	0.26	0.10	0.00	0.02
32	0.02	0.24	0.01	0.14	0.59	0.00	0.00
33	0.01	0.25	0.00	0.44	0.00	0.00	0.29
34	0.45	0.32	0.00	0.18	0.05	0.00	0.00
35	0.12	0.08	0.00	0.63	0.16	0.00	0.00
36	0.10	0.15	0.08	0.31	0.11	0.00	0.24
37	0.00	0.07	0.06	0.81	0.05	0.00	0.00
38	0.00	0.23	0.04	0.65	0.07	0.00	0.02
39	0.01	0.11	0.06	0.71	0.10	0.00	0.00
40	0.00	0.15	0.11	0.36	0.00	0.37	0.00
41	0.00	0.30	0.00	0.37	0.08	0.24	0.00
42	0.29	0.15	0.31	0.19	0.04	0.00	0.02
43	0.03	0.37	0.36	0.16	0.08	0.00	0.00
44	0.02	0.11	0.69	0.09	0.06	0.01	0.01
45	0.00	0.08	0.73	0.08	0.09	0.00	0.02
46	0.02	0.19	0.08	0.55	0.15	0.00	0.00
47	0.02	0.10	0.26	0.24	0.37	0.00	0.02
48	0.00	0.04	0.70	0.22	0.00	0.00	0.03
49	0.00	0.00	0.11	0.74	0.14	0.00	0.02
50	0.00	0.87	0.05	0.00	0.00	0.00	0.07
51	0.00	0.01	0.01	0.86	0.10	0.00	0.01
52	0.00	0.00	0.08	0.83	0.03	0.00	0.05

Word Cloud of Model 1



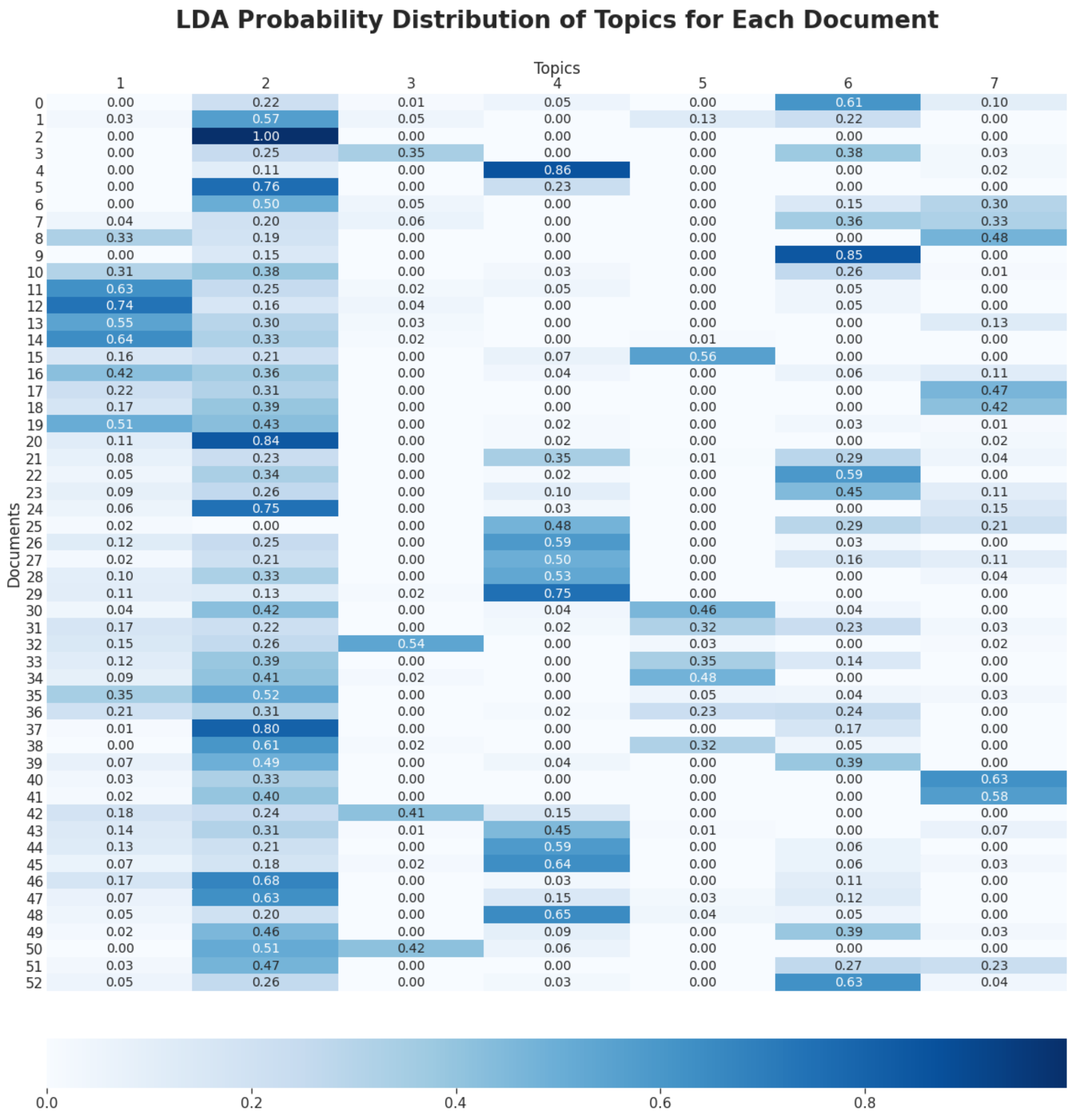
Bar Graph of Model 1



Appendix III

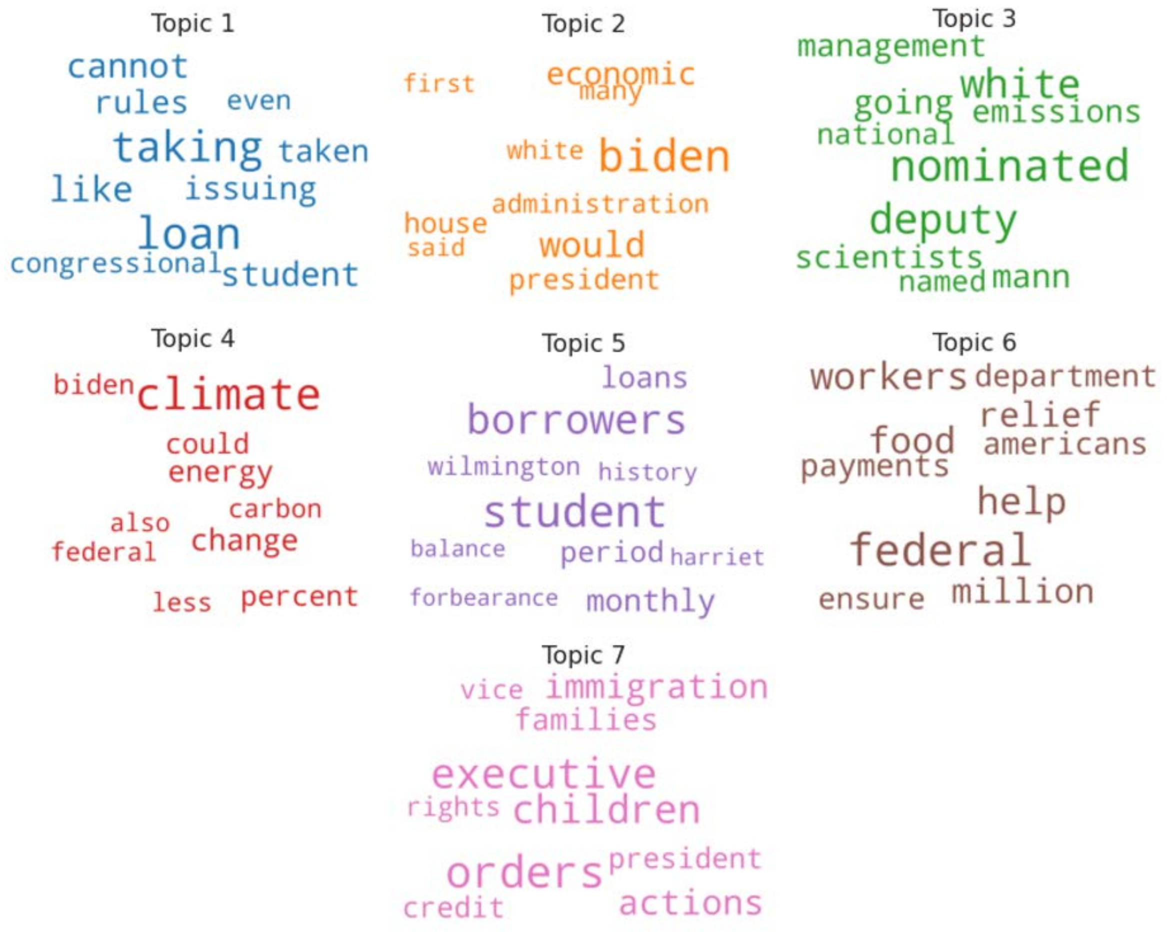
Model Two Results

Heat Map of Model 2

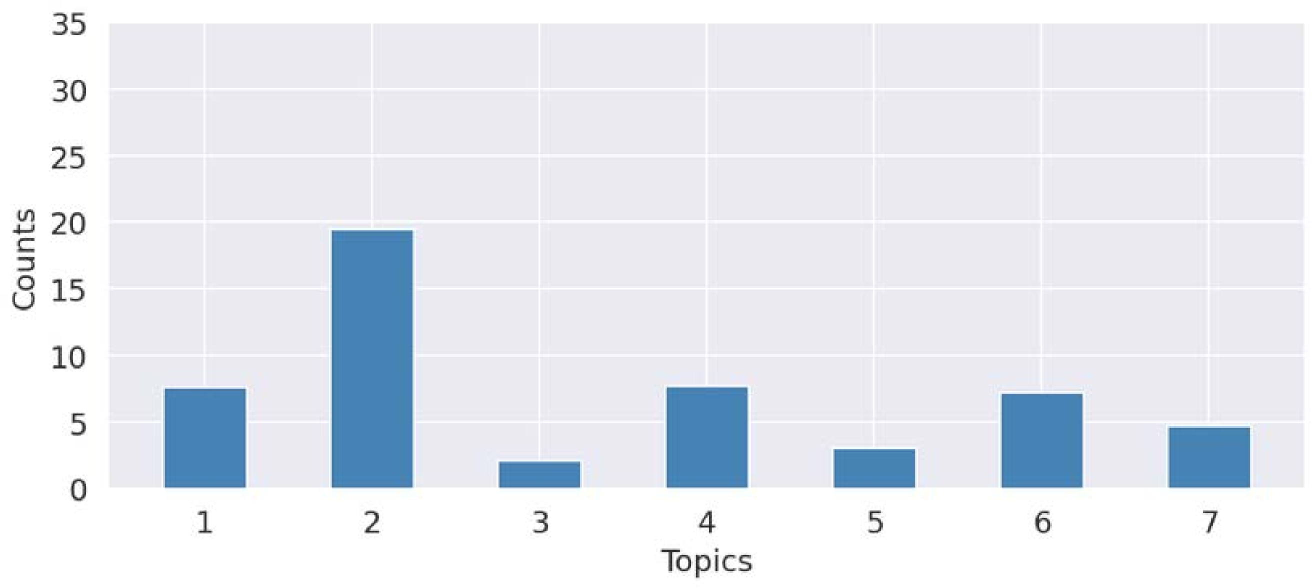


Documents	1	2	3	4	5	6	7
0	0.00	0.22	0.01	0.05	0.00	0.61	0.10
1	0.03	0.57	0.05	0.00	0.13	0.22	0.00
2	0.00	1.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.25	0.35	0.00	0.00	0.38	0.03
4	0.00	0.11	0.00	0.86	0.00	0.00	0.02
5	0.00	0.76	0.00	0.23	0.00	0.00	0.00
6	0.00	0.50	0.05	0.00	0.00	0.15	0.30
7	0.04	0.20	0.06	0.00	0.00	0.36	0.33
8	0.33	0.19	0.00	0.00	0.00	0.00	0.48
9	0.00	0.15	0.00	0.00	0.00	0.85	0.00
10	0.31	0.38	0.00	0.03	0.00	0.26	0.01
11	0.63	0.25	0.02	0.05	0.00	0.05	0.00
12	0.74	0.16	0.04	0.00	0.00	0.05	0.00
13	0.55	0.30	0.03	0.00	0.00	0.00	0.13
14	0.64	0.33	0.02	0.00	0.01	0.00	0.00
15	0.16	0.21	0.00	0.07	0.56	0.00	0.00
16	0.42	0.36	0.00	0.04	0.00	0.06	0.11
17	0.22	0.31	0.00	0.00	0.00	0.00	0.47
18	0.17	0.39	0.00	0.00	0.00	0.00	0.42
19	0.51	0.43	0.00	0.02	0.00	0.03	0.01
20	0.11	0.84	0.00	0.02	0.00	0.00	0.02
21	0.08	0.23	0.00	0.35	0.01	0.29	0.04
22	0.05	0.34	0.00	0.02	0.00	0.59	0.00
23	0.09	0.26	0.00	0.10	0.00	0.45	0.11
24	0.06	0.75	0.00	0.03	0.00	0.00	0.15
25	0.02	0.00	0.00	0.48	0.00	0.29	0.21
26	0.12	0.25	0.00	0.59	0.00	0.03	0.00
27	0.02	0.21	0.00	0.50	0.00	0.16	0.11
28	0.10	0.33	0.00	0.53	0.00	0.00	0.04
29	0.11	0.13	0.02	0.75	0.00	0.00	0.00
30	0.04	0.42	0.00	0.04	0.46	0.04	0.00
31	0.17	0.22	0.00	0.02	0.32	0.23	0.03
32	0.15	0.26	0.54	0.00	0.03	0.00	0.02
33	0.12	0.39	0.00	0.00	0.35	0.14	0.00
34	0.09	0.41	0.02	0.00	0.48	0.00	0.00
35	0.35	0.52	0.00	0.00	0.05	0.04	0.03
36	0.21	0.31	0.00	0.02	0.23	0.24	0.00
37	0.01	0.80	0.00	0.00	0.00	0.17	0.00
38	0.00	0.61	0.02	0.00	0.32	0.05	0.00
39	0.07	0.49	0.00	0.04	0.00	0.39	0.00
40	0.03	0.33	0.00	0.00	0.00	0.00	0.63
41	0.02	0.40	0.00	0.00	0.00	0.00	0.58
42	0.18	0.24	0.41	0.15	0.00	0.00	0.00
43	0.14	0.31	0.01	0.45	0.01	0.00	0.07
44	0.13	0.21	0.00	0.59	0.00	0.06	0.00
45	0.07	0.18	0.02	0.64	0.00	0.06	0.03
46	0.17	0.68	0.00	0.03	0.00	0.11	0.00
47	0.07	0.63	0.00	0.15	0.03	0.12	0.00
48	0.05	0.20	0.00	0.65	0.04	0.05	0.00
49	0.02	0.46	0.00	0.09	0.00	0.39	0.03
50	0.00	0.51	0.42	0.06	0.00	0.00	0.00
51	0.03	0.47	0.00	0.00	0.00	0.27	0.23
52	0.05	0.26	0.00	0.03	0.00	0.63	0.04

Word Cloud of Model 2



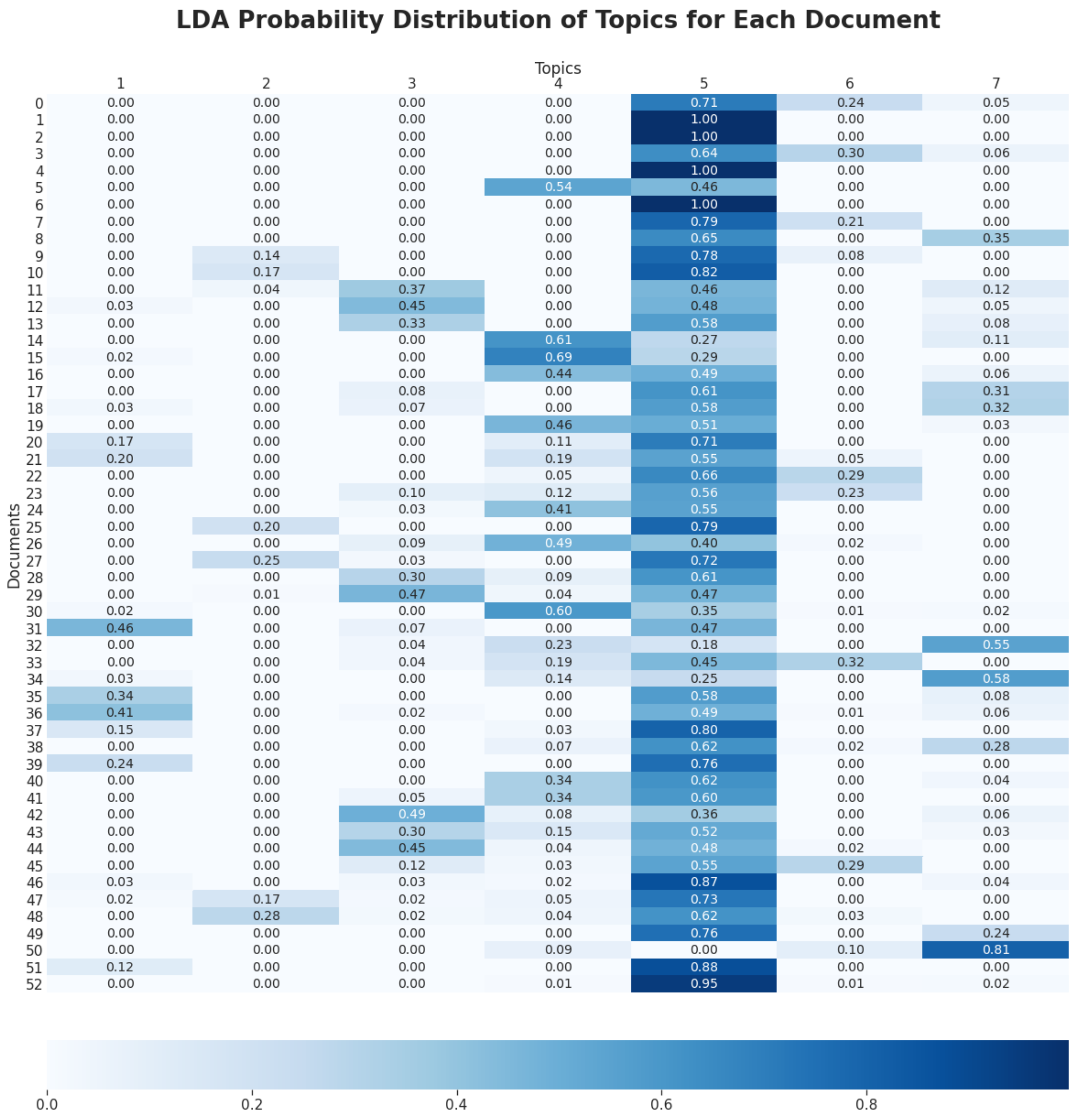
Bar Graph of Model 2



Appendix IV

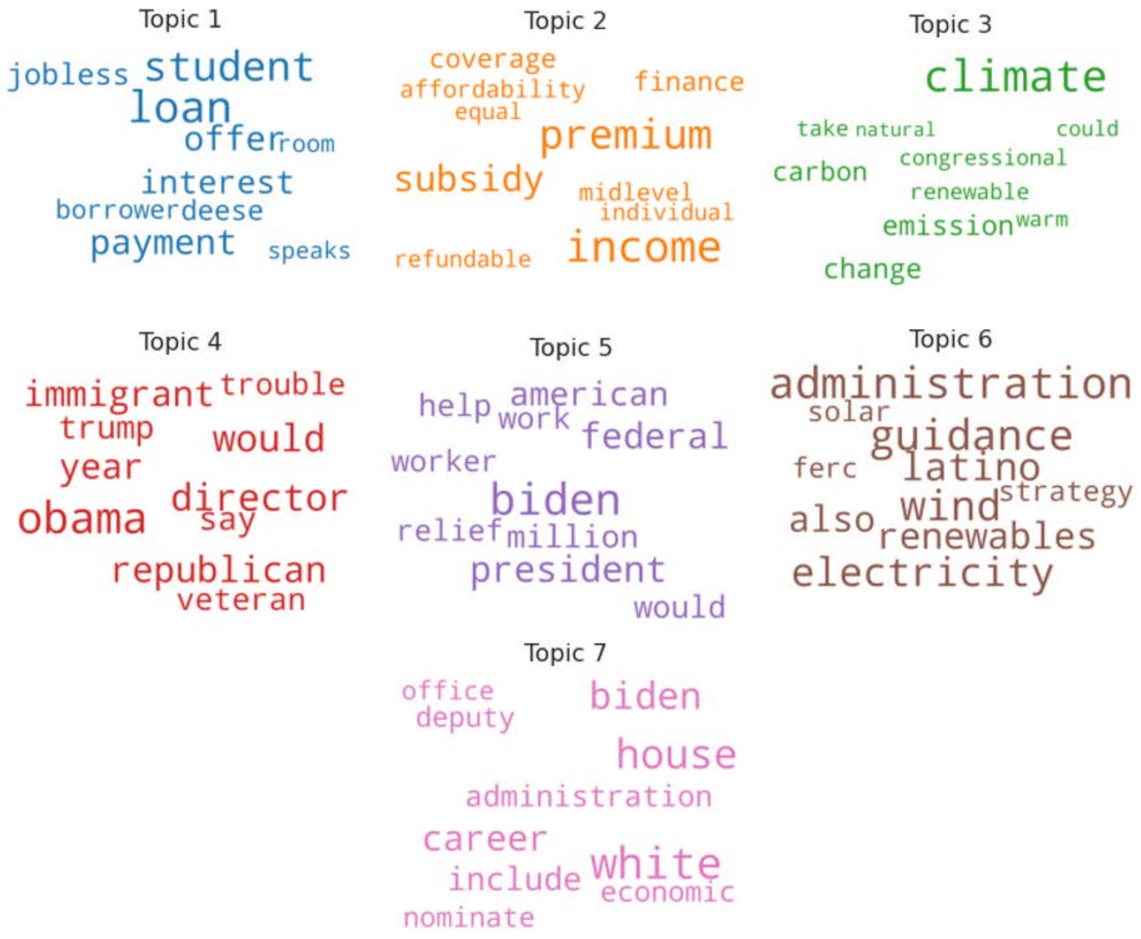
Model Three Results

Heat Map of Model 3

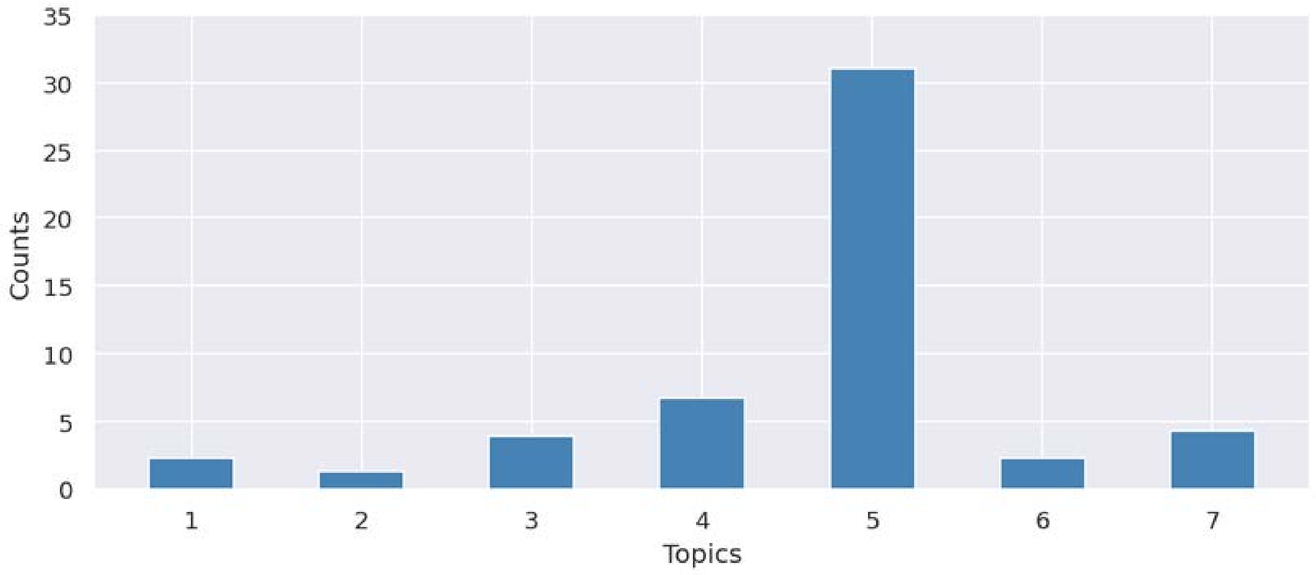


Documents	1	2	3	4	5	6	7
0	0.00	0.00	0.00	0.00	0.71	0.24	0.05
1	0.00	0.00	0.00	0.00	1.00	0.00	0.00
2	0.00	0.00	0.00	0.00	1.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.64	0.30	0.06
4	0.00	0.00	0.00	0.00	1.00	0.00	0.00
5	0.00	0.00	0.00	0.54	0.46	0.00	0.00
6	0.00	0.00	0.00	0.00	1.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.79	0.21	0.00
8	0.00	0.00	0.00	0.00	0.65	0.00	0.35
9	0.00	0.14	0.00	0.00	0.78	0.08	0.00
10	0.00	0.17	0.00	0.00	0.82	0.00	0.00
11	0.00	0.04	0.37	0.00	0.46	0.00	0.12
12	0.03	0.00	0.45	0.00	0.48	0.00	0.05
13	0.00	0.00	0.33	0.00	0.58	0.00	0.08
14	0.00	0.00	0.00	0.61	0.27	0.00	0.11
15	0.02	0.00	0.00	0.69	0.29	0.00	0.00
16	0.00	0.00	0.00	0.44	0.49	0.00	0.06
17	0.00	0.00	0.08	0.00	0.61	0.00	0.31
18	0.03	0.00	0.07	0.00	0.58	0.00	0.32
19	0.00	0.00	0.00	0.46	0.51	0.00	0.03
20	0.17	0.00	0.00	0.11	0.71	0.00	0.00
21	0.20	0.00	0.00	0.19	0.55	0.05	0.00
22	0.00	0.00	0.00	0.05	0.66	0.29	0.00
23	0.00	0.00	0.10	0.12	0.56	0.23	0.00
24	0.00	0.00	0.03	0.41	0.55	0.00	0.00
25	0.00	0.20	0.00	0.00	0.79	0.00	0.00
26	0.00	0.00	0.09	0.49	0.40	0.02	0.00
27	0.00	0.25	0.03	0.00	0.72	0.00	0.00
28	0.00	0.00	0.30	0.09	0.61	0.00	0.00
29	0.00	0.01	0.47	0.04	0.47	0.00	0.00
30	0.02	0.00	0.00	0.60	0.35	0.01	0.02
31	0.46	0.00	0.07	0.00	0.47	0.00	0.00
32	0.00	0.00	0.04	0.23	0.18	0.00	0.55
33	0.00	0.00	0.04	0.19	0.45	0.32	0.00
34	0.03	0.00	0.00	0.14	0.25	0.00	0.58
35	0.34	0.00	0.00	0.00	0.58	0.00	0.08
36	0.41	0.00	0.02	0.00	0.49	0.01	0.06
37	0.15	0.00	0.00	0.03	0.80	0.00	0.00
38	0.00	0.00	0.00	0.07	0.62	0.02	0.28
39	0.24	0.00	0.00	0.00	0.76	0.00	0.00
40	0.00	0.00	0.00	0.34	0.62	0.00	0.04
41	0.00	0.00	0.05	0.34	0.60	0.00	0.00
42	0.00	0.00	0.49	0.08	0.36	0.00	0.06
43	0.00	0.00	0.30	0.15	0.52	0.00	0.03
44	0.00	0.00	0.45	0.04	0.48	0.02	0.00
45	0.00	0.00	0.12	0.03	0.55	0.29	0.00
46	0.03	0.00	0.03	0.02	0.87	0.00	0.04
47	0.02	0.17	0.02	0.05	0.73	0.00	0.00
48	0.00	0.28	0.02	0.04	0.62	0.03	0.00
49	0.00	0.00	0.00	0.00	0.76	0.00	0.24
50	0.00	0.00	0.00	0.09	0.00	0.10	0.81
51	0.12	0.00	0.00	0.00	0.88	0.00	0.00
52	0.00	0.00	0.00	0.01	0.95	0.01	0.02

Word Cloud of Model 3



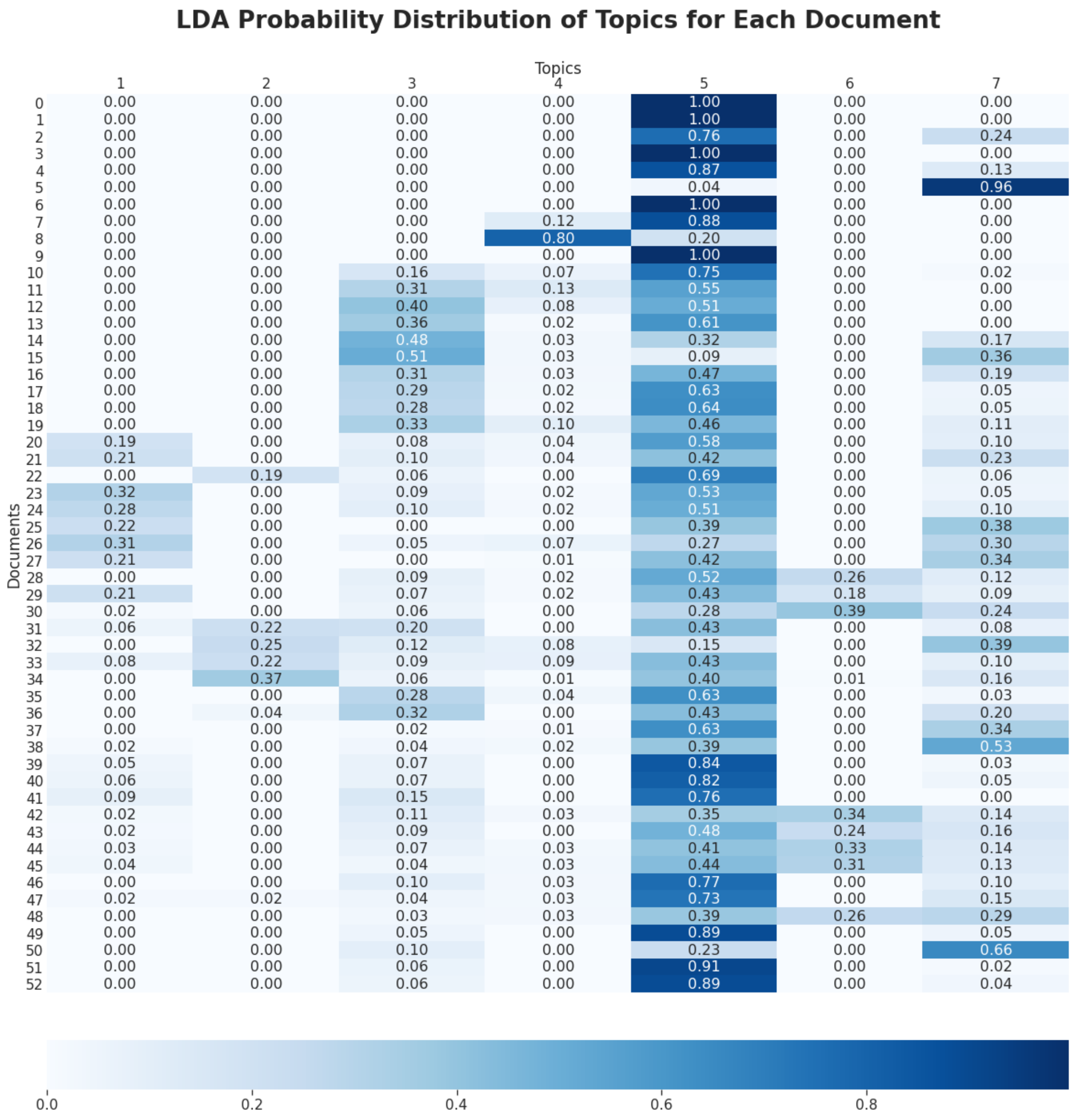
Bar Graph of Model 3



Appendix V

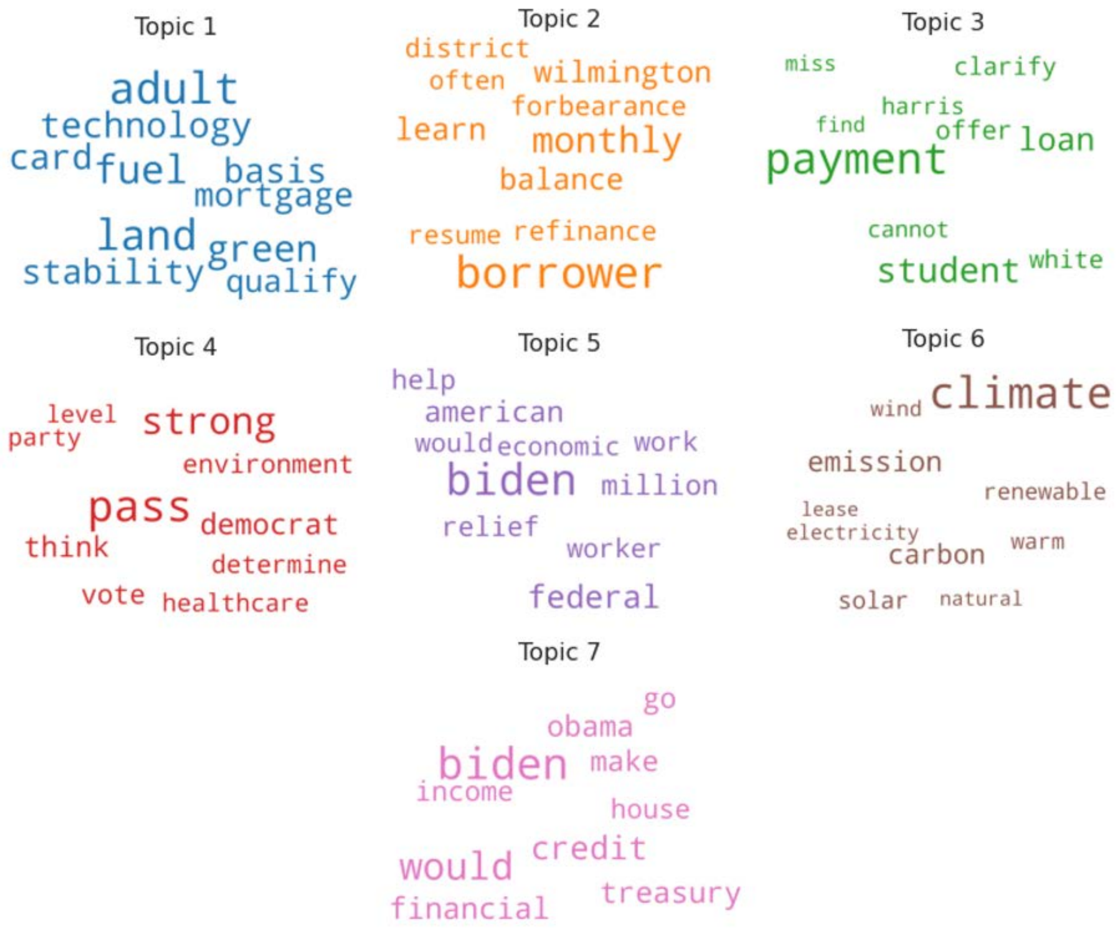
Model Four Results

Heat Map of Model 4

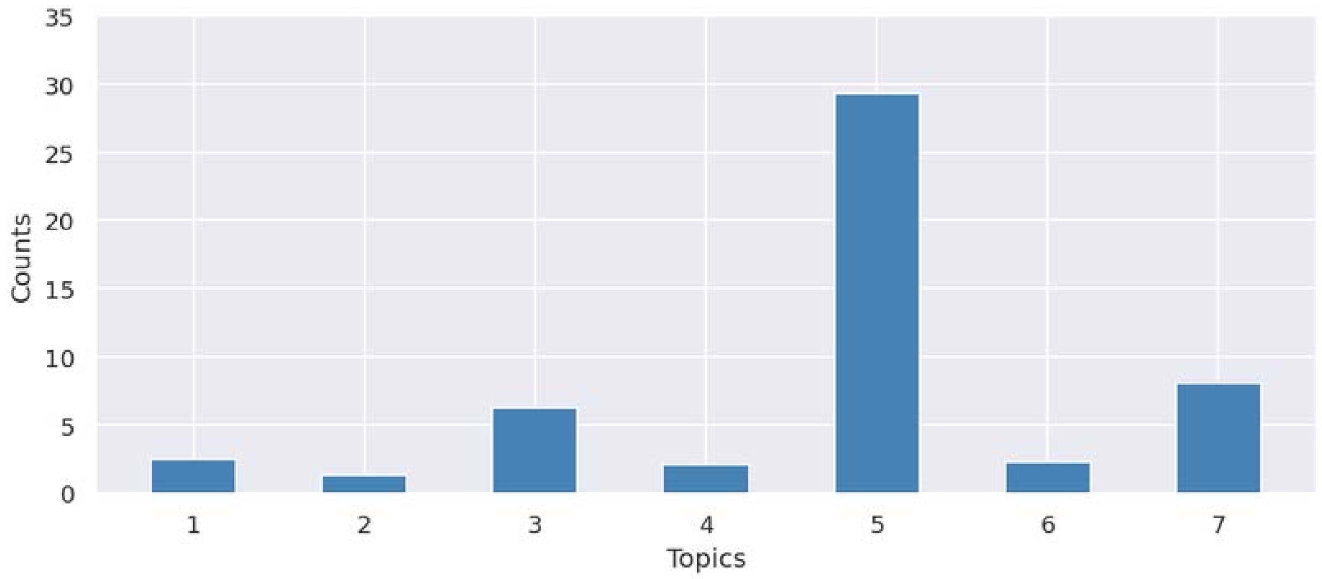


Documents	1	2	3	4	5	6	7
0	0.00	0.00	0.00	0.00	1.00	0.00	0.00
1	0.00	0.00	0.00	0.00	1.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.76	0.00	0.24
3	0.00	0.00	0.00	0.00	1.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.87	0.00	0.13
5	0.00	0.00	0.00	0.00	0.04	0.00	0.96
6	0.00	0.00	0.00	0.00	1.00	0.00	0.00
7	0.00	0.00	0.00	0.12	0.88	0.00	0.00
8	0.00	0.00	0.00	0.80	0.20	0.00	0.00
9	0.00	0.00	0.00	0.00	1.00	0.00	0.00
10	0.00	0.00	0.16	0.07	0.75	0.00	0.02
11	0.00	0.00	0.31	0.13	0.55	0.00	0.00
12	0.00	0.00	0.40	0.08	0.51	0.00	0.00
13	0.00	0.00	0.36	0.02	0.61	0.00	0.00
14	0.00	0.00	0.48	0.03	0.32	0.00	0.17
15	0.00	0.00	0.51	0.03	0.09	0.00	0.36
16	0.00	0.00	0.31	0.03	0.47	0.00	0.19
17	0.00	0.00	0.29	0.02	0.63	0.00	0.05
18	0.00	0.00	0.28	0.02	0.64	0.00	0.05
19	0.00	0.00	0.33	0.10	0.46	0.00	0.11
20	0.19	0.00	0.08	0.04	0.58	0.00	0.10
21	0.21	0.00	0.10	0.04	0.42	0.00	0.23
22	0.00	0.19	0.06	0.00	0.69	0.00	0.06
23	0.32	0.00	0.09	0.02	0.53	0.00	0.05
24	0.28	0.00	0.10	0.02	0.51	0.00	0.10
25	0.22	0.00	0.00	0.00	0.39	0.00	0.38
26	0.31	0.00	0.05	0.07	0.27	0.00	0.30
27	0.21	0.00	0.00	0.01	0.42	0.00	0.34
28	0.00	0.00	0.09	0.02	0.52	0.26	0.12
29	0.21	0.00	0.07	0.02	0.43	0.18	0.09
30	0.02	0.00	0.06	0.00	0.28	0.39	0.24
31	0.06	0.22	0.20	0.00	0.43	0.00	0.08
32	0.00	0.25	0.12	0.08	0.15	0.00	0.39
33	0.08	0.22	0.09	0.09	0.43	0.00	0.10
34	0.00	0.37	0.06	0.01	0.40	0.01	0.16
35	0.00	0.00	0.28	0.04	0.63	0.00	0.03
36	0.00	0.04	0.32	0.00	0.43	0.00	0.20
37	0.00	0.00	0.02	0.01	0.63	0.00	0.34
38	0.02	0.00	0.04	0.02	0.39	0.00	0.53
39	0.05	0.00	0.07	0.00	0.84	0.00	0.03
40	0.06	0.00	0.07	0.00	0.82	0.00	0.05
41	0.09	0.00	0.15	0.00	0.76	0.00	0.00
42	0.02	0.00	0.11	0.03	0.35	0.34	0.14
43	0.02	0.00	0.09	0.00	0.48	0.24	0.16
44	0.03	0.00	0.07	0.03	0.41	0.33	0.14
45	0.04	0.00	0.04	0.03	0.44	0.31	0.13
46	0.00	0.00	0.10	0.03	0.77	0.00	0.10
47	0.02	0.02	0.04	0.03	0.73	0.00	0.15
48	0.00	0.00	0.03	0.03	0.39	0.26	0.29
49	0.00	0.00	0.05	0.00	0.89	0.00	0.05
50	0.00	0.00	0.10	0.00	0.23	0.00	0.66
51	0.00	0.00	0.06	0.00	0.91	0.00	0.02
52	0.00	0.00	0.06	0.00	0.89	0.00	0.04

Word Cloud of Model 4



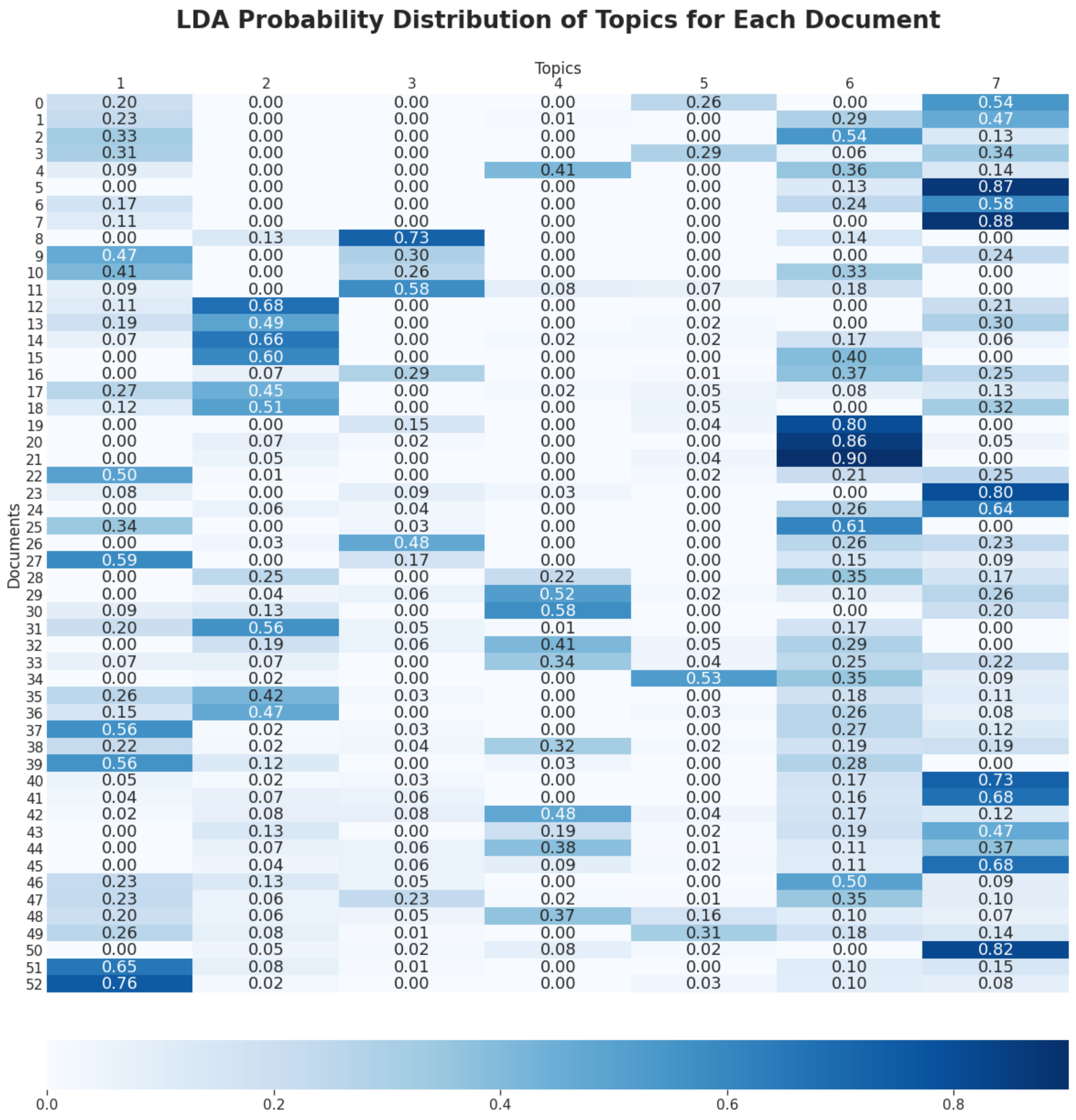
Bar Graph of Model 4



Appendix VI

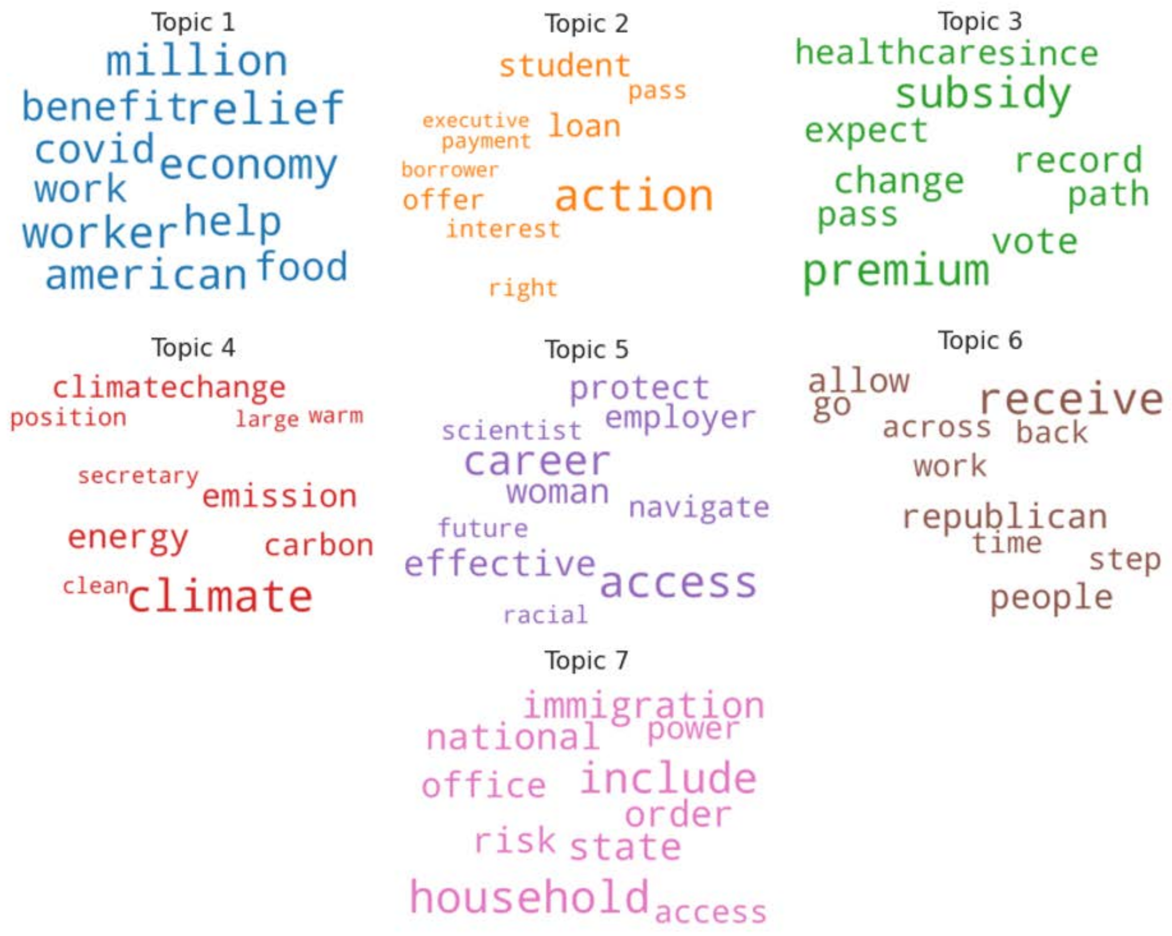
Model Five Results

Heat Map of Model 5

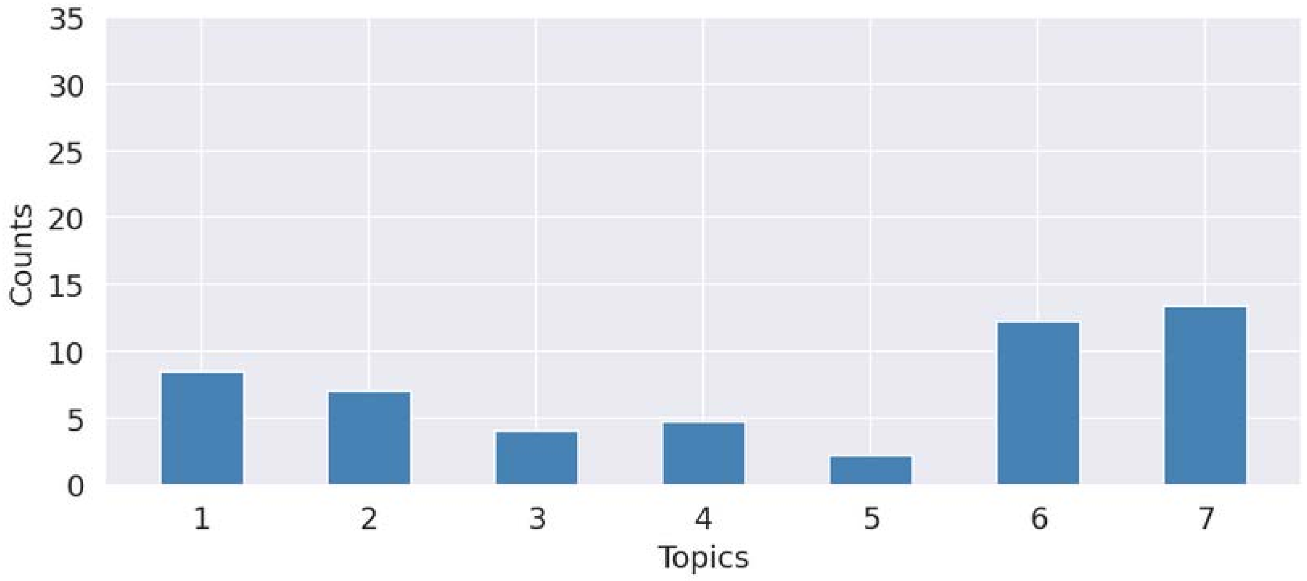


Documents	1	2	3	4	5	6	7
0	0.20	0.00	0.00	0.00	0.26	0.00	0.54
1	0.23	0.00	0.00	0.01	0.00	0.29	0.47
2	0.33	0.00	0.00	0.00	0.00	0.54	0.13
3	0.31	0.00	0.00	0.00	0.29	0.06	0.34
4	0.09	0.00	0.00	0.41	0.00	0.36	0.14
5	0.00	0.00	0.00	0.00	0.00	0.13	0.87
6	0.17	0.00	0.00	0.00	0.00	0.24	0.58
7	0.11	0.00	0.00	0.00	0.00	0.00	0.88
8	0.00	0.13	0.73	0.00	0.00	0.14	0.00
9	0.47	0.00	0.30	0.00	0.00	0.00	0.24
10	0.41	0.00	0.26	0.00	0.00	0.33	0.00
11	0.09	0.00	0.58	0.08	0.07	0.18	0.00
12	0.11	0.68	0.00	0.00	0.00	0.00	0.21
13	0.19	0.49	0.00	0.00	0.02	0.00	0.30
14	0.07	0.66	0.00	0.02	0.02	0.17	0.06
15	0.00	0.60	0.00	0.00	0.00	0.40	0.00
16	0.00	0.07	0.29	0.00	0.01	0.37	0.25
17	0.27	0.45	0.00	0.02	0.05	0.08	0.13
18	0.12	0.51	0.00	0.00	0.05	0.00	0.32
19	0.00	0.00	0.15	0.00	0.04	0.80	0.00
20	0.00	0.07	0.02	0.00	0.00	0.86	0.05
21	0.00	0.05	0.00	0.00	0.04	0.90	0.00
22	0.50	0.01	0.00	0.00	0.02	0.21	0.25
23	0.08	0.00	0.09	0.03	0.00	0.00	0.80
24	0.00	0.06	0.04	0.00	0.00	0.26	0.64
25	0.34	0.00	0.03	0.00	0.00	0.61	0.00
26	0.00	0.03	0.48	0.00	0.00	0.26	0.23
27	0.59	0.00	0.17	0.00	0.00	0.15	0.09
28	0.00	0.25	0.00	0.22	0.00	0.35	0.17
29	0.00	0.04	0.06	0.52	0.02	0.10	0.26
30	0.09	0.13	0.00	0.58	0.00	0.00	0.20
31	0.20	0.56	0.05	0.01	0.00	0.17	0.00
32	0.00	0.19	0.06	0.41	0.05	0.29	0.00
33	0.07	0.07	0.00	0.34	0.04	0.25	0.22
34	0.00	0.02	0.00	0.00	0.53	0.35	0.09
35	0.26	0.42	0.03	0.00	0.00	0.18	0.11
36	0.15	0.47	0.00	0.00	0.03	0.26	0.08
37	0.56	0.02	0.03	0.00	0.00	0.27	0.12
38	0.22	0.02	0.04	0.32	0.02	0.19	0.19
39	0.56	0.12	0.00	0.03	0.00	0.28	0.00
40	0.05	0.02	0.03	0.00	0.00	0.17	0.73
41	0.04	0.07	0.06	0.00	0.00	0.16	0.68
42	0.02	0.08	0.08	0.48	0.04	0.17	0.12
43	0.00	0.13	0.00	0.19	0.02	0.19	0.47
44	0.00	0.07	0.06	0.38	0.01	0.11	0.37
45	0.00	0.04	0.06	0.09	0.02	0.11	0.68
46	0.23	0.13	0.05	0.00	0.00	0.50	0.09
47	0.23	0.06	0.23	0.02	0.01	0.35	0.10
48	0.20	0.06	0.05	0.37	0.16	0.10	0.07
49	0.26	0.08	0.01	0.00	0.31	0.18	0.14
50	0.00	0.05	0.02	0.08	0.02	0.00	0.82
51	0.65	0.08	0.01	0.00	0.00	0.10	0.15
52	0.76	0.02	0.00	0.00	0.03	0.10	0.08

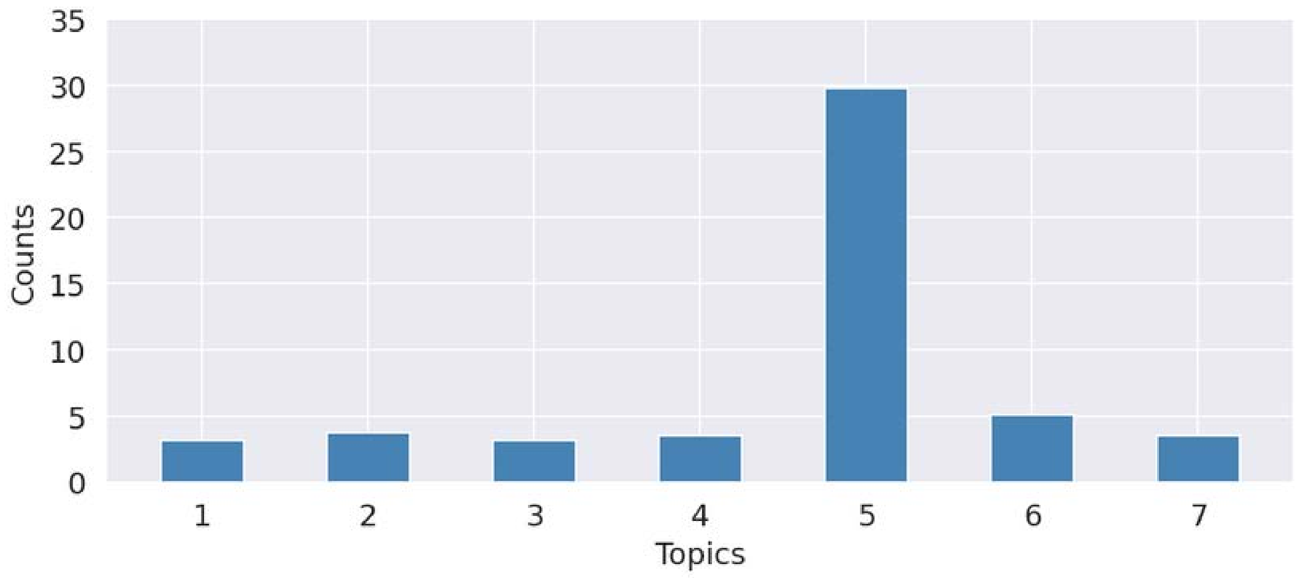
Word Cloud of Model 5



Bar Graph of Model 5



Excluding the word administration as a stop word


Counts
Topics