Topic Modeling for COVID-19 Research Papers

Brandon Le, Yves Wienecke, Angie McGraw, Tu Vu, Keaton Kraiger

COVID-19 has become one of the most impactful events in our history. The pandemic has significantly disrupted the manner in which people communicate and learn. One of the leading voices in the pandemic is the medical field, providing us their research and insights in these uncertain times. Natural language processing (NLP) will be utilized to help us understand the semantic content of research in the field of epidemiology, as well as the impacts that COVID-19 has had on research topics. Utilization of NLP will help highlight shifts in research topics leading up to and during the pandemic. To aid in NLP, topic modeling and topic clustering is explored. Topic modeling is used for unsupervised analysis of large amounts of data. In this research, a dataset with over 138,000 scholarly articles is analyzed using topic modeling techniques of Latent Dirichlet Allocation (LDA) and topic clustering with Word2vec.

1. Introduction

Topic modeling is a method for identifying a number of hidden topics and related keyphrases in a corpus. It can be used to provide an understanding of the spread of topics in the corpus and the similarity of these topics. Additionally, topic modelling can be used for other Natural Language Processing (NLP) tasks, such as information retrieval (Wei and Croft, 2006), search engines, word sense disambiguation (Chaplot and Salakhutdinov, 2018), and machine

translation. Since the goal of topic modelling is to condense large amounts of data into relatively small topics representative of the semantic content of the corpus, the results produced from topic modelling can be considered as a very succinct summary of the corpus. The amount of research conducted throughout the years has increasingly grown, and the magnitude of papers published by epidemiological journals has exploded in response to the current COVID-19 epidemic. It is difficult to understand and visualize the semantic content of an increasingly growing set of research, and the impact that a worldwide pandemic may have on this content.

In order to better understand and visualize the semantic content of research, our group used the gensim implementations of LDA and word embeddings (Word2vec) to analyze the semantic content of research papers in the field of epidemiology, as collected by the Allen Institute Open Research Dataset. A different approach to understanding the semantic content of a corpus is topic clustering (Lee et al. 2012). However, topic clustering and topic modeling may be sensitive to noise and improperly cleaned data, which places a particular importance on the preprocessing step of the NLP pipeline. Initially, we believed that topic modeling and topic clustering were quite similar, however, our exploration into these two areas of semantic analysis led to the discovery these are two inherently different approaches. Topic modeling learns the latent topics that are associated with a corpus. Topic clustering, however, learns to cluster word embeddings into coherent groups -- topics. The algorithm we used, LDA, focuses on per-word document distributions and per-topic document distributions (Vulić et al. 2012). The initial focus of this research paper was in topic modelling rather than topic clustering, but it turns out that topic modelling is not possible with just word embeddings. To further our analysis of the Allen

Institute dataset, our main focus moved towards LDA, but we also included the work we did to analyze the semantic content of the dataset with Word2vec. The overall purpose of this research is to better understand the current state of the research done in the medical field with regards to COVID19. A condensed form of the methods used in this research is as follows: (1) import the dataset, (2) preprocess the text, (3) create the gensim dictionary and corpus, (4) build and train the LDA or Word2vec model, and (5) analyze the output of the models.

1.1 LDA and Word Embeddings (Word2vec)

LDA is one of the most popular topic modeling analysis mechanisms, developed in 2003 by David Blei, Andrew Ng, and Michael I. Jordan (Blei et al. 2003). LDA is a probabilistic model used for analyzing discrete data (Blei et al. 2003). It improves upon Latent Semantic Analysis/Indexing (LSA/LSI) and probabilistic LSA/LSI (pLSA/pLSI) by introducing two dirichlet priors to learn the probability distributions of words to topics and topics to documents. It is a three-level hierarchical Bayesian model, where each item in the collection is modeled, in relation to the different topics spread over the dataset (Blei et al. 2003).

In 2013, Tomas Mikolov developed Word2vec (Goldberg and Levy, 2014). The model provides word embeddings (Goldberg and Levy, 2014). Word2vec produces vector representations of words, allowing users to analyze the words' semantic meanings. Word2vec has proven to be useful in carrying out various NLP tasks (Rong, 2014). Word2vec is a method for creating continuous vector representations of words from a text corpus. This process of "vectorizing" words belongs to a family of techniques for creating word embeddings where

learned representations of words are mapped onto a vector space. Word2vec creates a dense distributed representation for each word in a vocabulary, hopefully capturing semantic and syntactic relations between words.

LDA and Word2vec have a couple of differences, other than their relationship to topic modeling and topic clustering. First, LDA creates document and topic representations that are more interpretable to humans than Word2vec. Second, Both LDA and Word2vec work with sets of documents, but LDA treats the sets as they are, whereas Word2vec treats them as a long text string (Nikita, 2016). Third, LDA predicts globally and Word2vec predicts locally. Finally, LDA predicts the keywords based on the entire corpus and Word2vec predicts related words from a given word. To summarize, LDA models document-to-word relationships and Word2vec models word-to-word relationships (Nikita 2016). Doc2vec, an improvement of the Word2vec model, performs better by including document distributions along with word distributions, but it is not in the scope of this research project.

1.2 Dataset

Our corpus consists of over 138,000 scholarly articles; 69,000 with full text, relating to COVID-19, SARS-CoV-2, and other related coronaviruses (Allen Institute for AI). Some of these articles are not in English, but detecting and removing them are out of scope of this research project. The COVID-19 Open Research Dataset was developed by the White House and a coalition of research groups (Allen Institute for AI). The main sources for the research papers are from Medline and PubMed Central (PMC). These research papers were gathered from

different websites; 31% from Medline, 18% from PMC, and 51% from other sites. For the scope of this project and the sake of processing time, we only worked with the abstracts. Each abstract is roughly a paragraph or two long. Figure [1] depicts the distribution of abstracts per year, spanning 62 years in total. These papers were published mainly in 2015 and 2020; 11% of papers are from 2020, 3% of papers are from 2015, and 86% were from other years or had no recorded publication date. The representation of documents from a certain year or field may impact the skew of topics. As seen in Figure [1], the top five journals are PLoS One, Journal of Virology, Virology Journal, bioRxiv, and Surgical Endoscopy. These journals primarily focus on epidemiology.

2 Methods

2.1 Preprocessing

The first step in the Natural Language Processing (NLP) pipeline involved obtaining the dataset through Kaggle (Kaggle, 2020). This research focused only on the *'metadata.csv*' file and further narrowed the scope of the data to only the following columns: 'publish_time', 'journal', and 'abstract'. The pandas and numpy library were used to isolate the abstracts from the dataset and represent the corpus as a vector of abstracts, with each document being an abstract in the form of a unicode string.

Following the acquisition of the dataset was the preprocessing step, whereby each document was transformed from a unicode string into a list of candidate keyphrases. First, each document was parsed and broken down into a list of word and punctuation tokens through the

Natural Language Toolkit (nltk) tokenizer implementation. Next, each token was morphologically normalized with the nltk implementation of the Porter Stemmer. Any tokens that were shorter than three characters were removed, as well as any tokens that did not contain at least one character in the ISO basic latin alphabet. The filtered tokens were then converted into lowercase. Bigram and trigram tokens were added to each document with the gensim collocation detector (Phrases). Finally, any tokens that contained stop words were removed. The resulting corpus after preprocessing was made up of documents as lists of stemmed alphanumeric unigrams, bigrams, and trigrams that did not contain stop words.

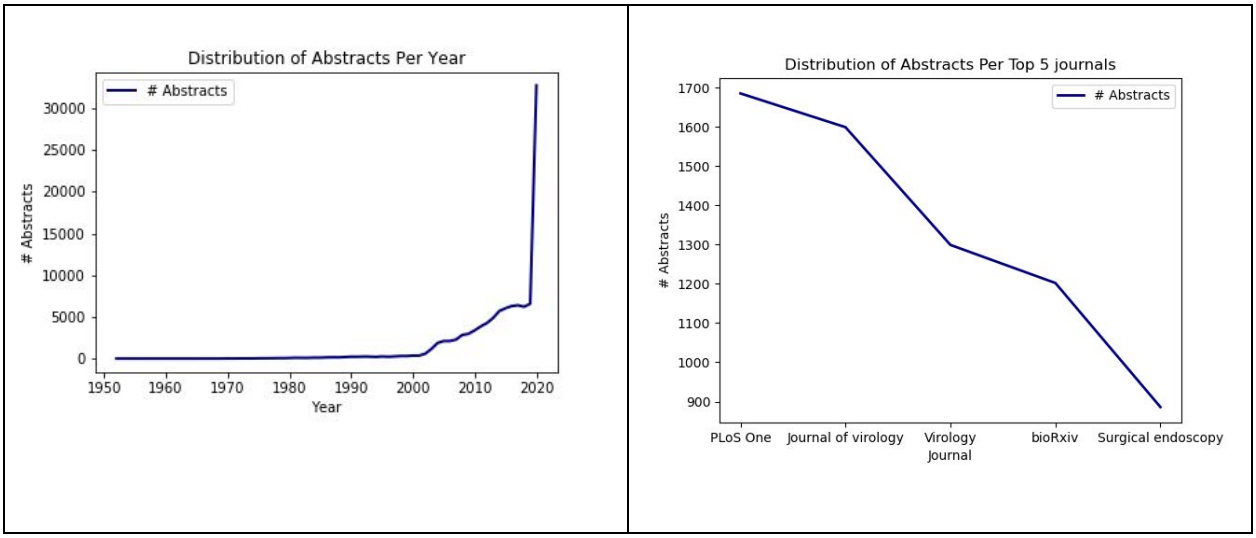

Figure [1]: The distribution of abstracts per year and per top 5 journals.

2.2 Latent Dirichlet Allocation (LDA)

A dictionary of the preprocessed corpus was generated with the gensim dictionary implementation, which associated each unique token with a unique identifier and a frequency count. Tokens which appeared in less than 100 documents and in over 50% of the documents in

the corpus were ignored from the dictionary. To prepare the corpus in a format readable for gensim's implementation of LDA, the corpus was transformed into a Bag-of-Words representation by executing the gensim doc2bow function on each document.

As mentioned before, LDA is a method of topic modeling, which generates a frequency of topics based words from a set of documents. For this step of the pipeline, we utilized the gensim's implementation of the LDA model. Various preprocessing and postprocessing parameters. For preprocessing, we looked at the frequencies of words above or below a certain threshold, and we looked at the differences between stemming and lemmatization. Stemming cuts off the endings of words, while still keeping their meaning. Lemmatization cuts off words to their original root form. For post-processing, we varied the number of topics and we varied eta; the Dirichlet prior for topic-word distributions.

We implemented the LDA algorithm on the whole collection of articles after preprocessing. For preprocessing, we split the process into two main approaches: stemming or lemmatization. For both approaches, we used dropping missing values, making words lowercase, stopping, and tokenization. Then, we either ran the stemming or lemmatization. For the model training, we varied the number of topics and eta.

For topic modeling, perplexity and topic coherence are two measures of how well a model represents the data. Perplexity measures how well a probability model predicts a sample. It captures how well a model responds to new data; data it has not seen before. This is measured using the normalized log-likelihood of a held-out test (Kapadia, 2019). Low perplexity is an indication that the probability distribution does not predict the sample well. Perplexity is not

strongly correlated with human judgement (Pleplé, 2013). As for topic coherence, it is a measure that scores a single topic by measuring the degree of semantic similarity between high-scoring words in the topic (Kapadia, 2019). Coherence is a measure of how well a statement or facts support each other (Kapadia, 2019). For our project, we are utilizing the UMass measure. This measure computes the coherence of a topic as the sum of pairwise distributional similarity scores over the set of topic words (Stevens et al. 2012). The equation is as follows: Coherence = $\sum_{i<j} score(w_i, w_j)$, where w represents the words (Pleplé, 2013). The higher the coherence between words, the better the relationship between them. A high topic coherence indicates that these words may be related to one another.

2.3 Word Embeddings (Word2vec)

For training our Word2vec model, we began by preprocessing the data similar to the methods described in section 2.2 with a few distinct differences. As mentioned in a previous section, we cleaned the text data by removing special characters/stop words, lower casing each word in our vocab, and then tokenize the document such that each word is its own element in a list representing the document with the nltk library. However, previous attempts at creating word embeddings have used various methods for processing text data and there is discussion on how text corpuses should be normalized (Mikolov et al. 2013). Similar to the original Word2vec model introduced by Mikolav et. al, we did not include stemming or lemmatization on the text corpus.

We utilized the gensim Word2vec model function to prepare and train our Word2vec model. The parameter *min_count* was used to create different word embeddings of our corpus, where *min_count* denoted a threshold for the frequency words must occur to learn their vector representation. Moreover, like Mikolav et. al's original model, our Word2vec model subsampled frequent words where each word in the corpus was discarded with probability $P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$, where $f(w_i)$ is the frequency of word w_i and t is a predefined threshold (Maaten and Hinton, 2008). For our model, we let $t = .001$. For visualizing our word2vec model, we used t-Distributed Stochastic Neighbor Embedding (t-SNE), a technique for dimensionality reduction that aids in visualizing high-dimensional data by giving each datapoint a location in a two-dimensional space.

2.4 Post Processing

The impact of various preprocessing approaches on the average coherence and perplexity of LDA were analyzed. The approaches which resulted in the best scores were used for optimal topic modeling. First, nltk's implementation of a tokenizer was chosen rather than parsing out words that matched a simple regex expression. The former kept punctuation as tokens, whereas the latter did not, resulting in a loss of syntactical and grammatical information necessary for accurate lemmatization. Second, stemming was chosen for morphological normalization over no normalization and lemmatization. As seen in Figure [2], the performance of topic modelling with LDA was best with stemming and worst with no morphological normalization. The difference in

performance after stemming and after lemmatization is relatively small, but stemming was chosen due to its significantly quicker runtime performance.

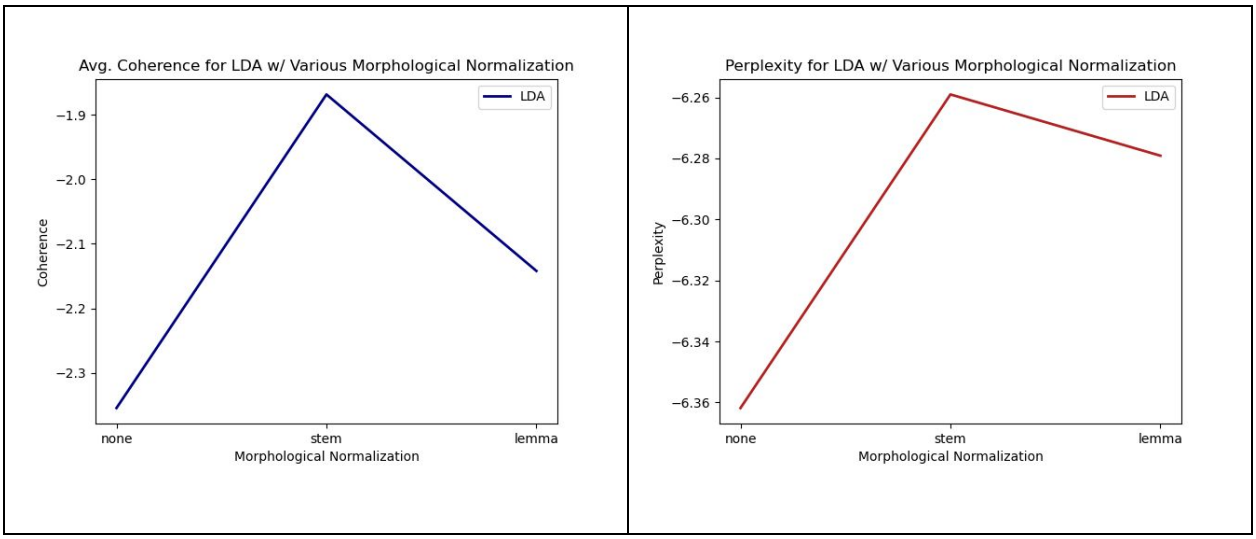

Figure [2]: Average coherence and perplexity scores for various methods of morphological normalization. Higher is better.

The no_above and no_below parameters were bounds for token frequency which filtered out tokens that either appeared in more than a certain percentage of documents in the corpus and tokens that did not appear in at least a certain amount of documents. The default value as set by gensim for no_above is 0.5, while the default value for no_below is 5. As seen in Figure [3], the value of no_above negatively correlated with the performance of LDA on our corpus. The performance of LDA improved by diminishing magnitudes as the value of no_below increased. The final chosen values for no_above and no_below was 0.5 and 100, respectively.

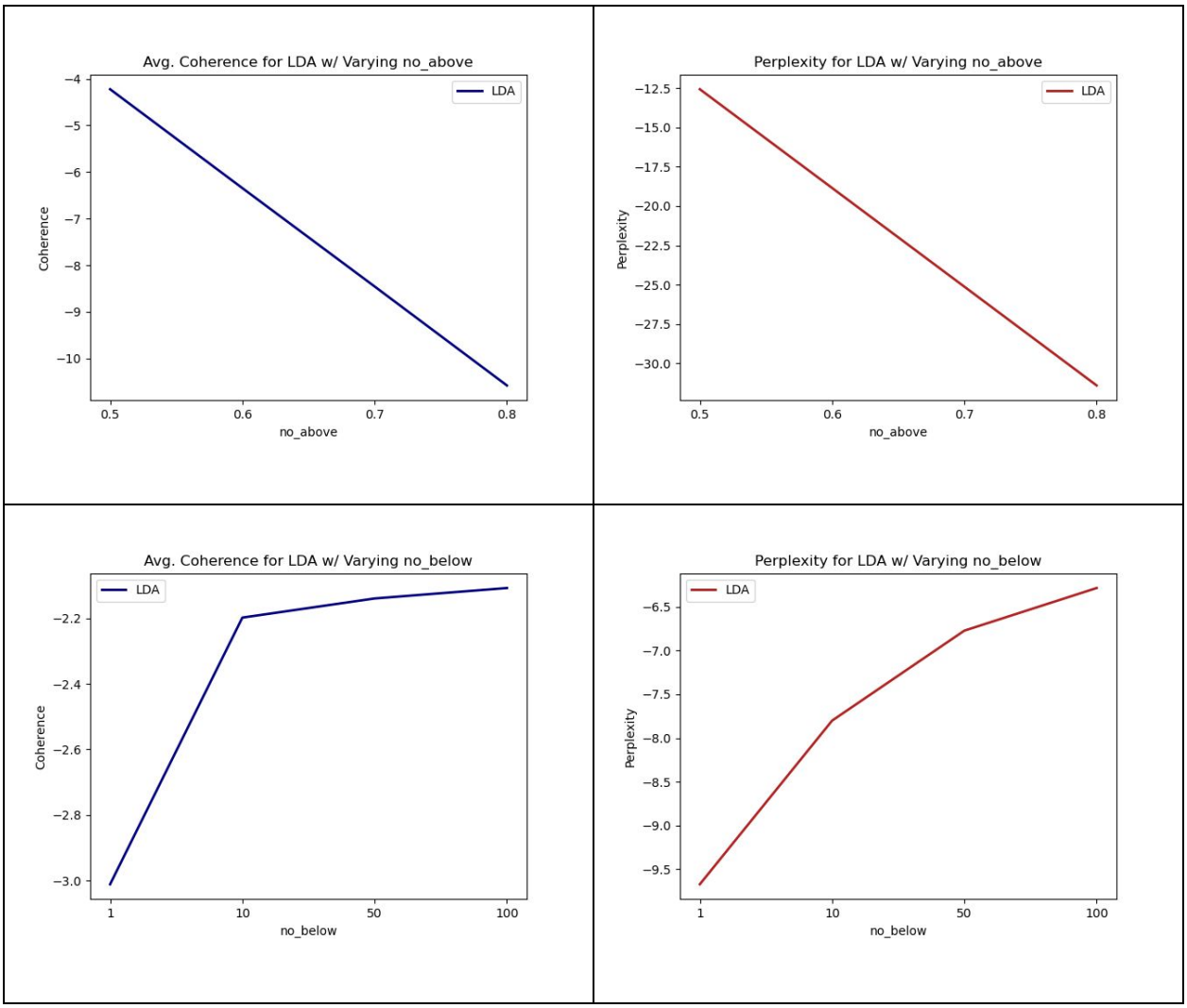

Figure [3] Average coherence and perplexity scores for various document frequency bounds. Higher is better. Gensim default is no_below=5; no_above=0.5.

Two hyperparameters for LDA were optimized, in a similar manner as the process for analyzing various preprocessing approaches. The first hyperparameter corresponds to the number of latent topics that LDA tries to model from the corpus. As seen in Figure [4], the performance of LDA on our corpus decreases as the number of topics increases, but this change is not

significant. For simplicity and runtime performance, the final value for num_topics was five topics. The second LDA hyperparameter was η, which is the dirichlet prior corresponding to the topic-word probability. A smaller value for η should result in more specific topics, while a larger value should generate broader topics. Various explicit symmetric values for η were analysed and compared to the 'auto' parameter, which learns asymmetric values for η. Figure [5] shows that the values set for eta did not have a large impact on the performance of LDA, but 'auto' was the best when considering both average coherence and perplexity.

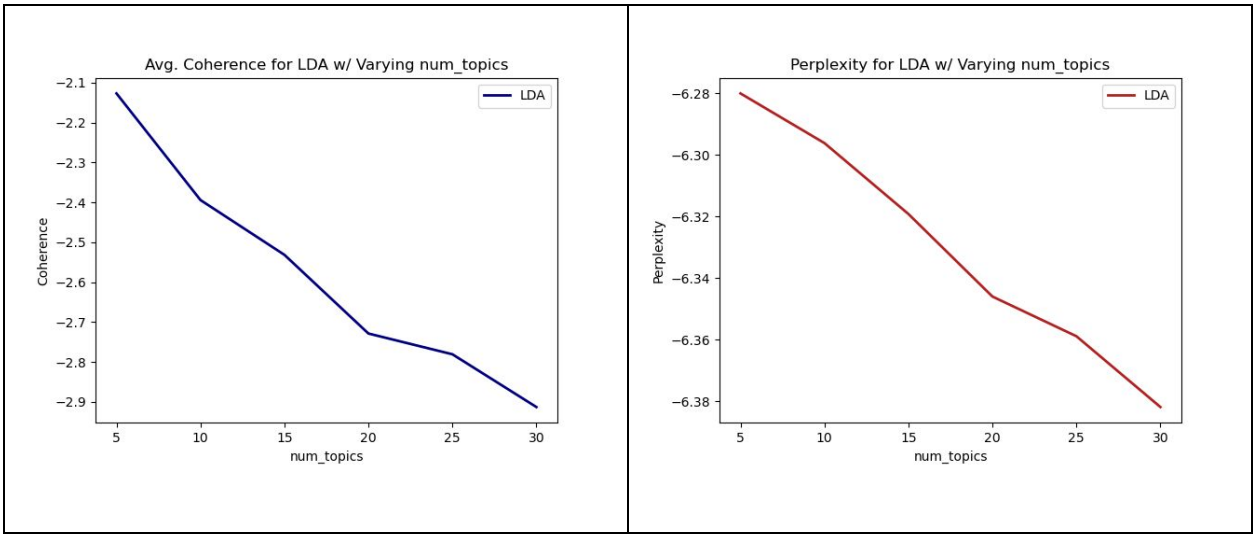

Figure [4]: Average coherence and perplexity scores for various amounts of latent topics. Higher is better.

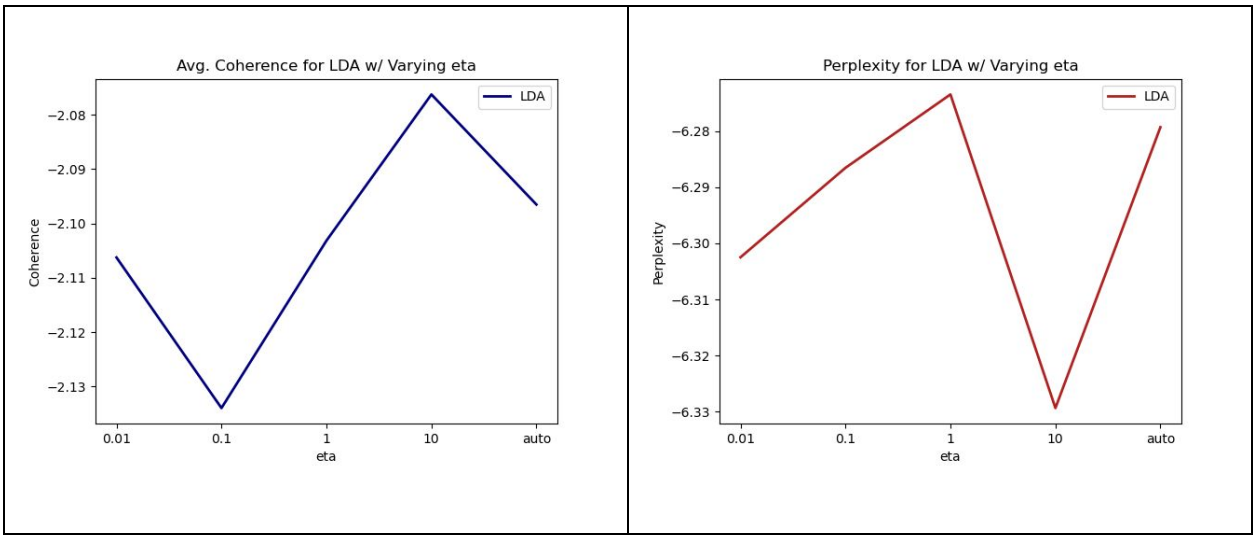

Figure [5]: Average coherence and perplexity scores for various values of the topic-word dirichlet prior; η. Higher is better.

3.1 Visualization techniques (LDA)

In order to visualize the results of LDA, pyLDAvis was used. pyLDAvis is a python library used for visualization of topic models through PCA (Wieringa, 2018). As seen in figure [6], there is a section that contains the Intertopic Distance Map (via multidimensional scaling) and the Top-30 Most Relevant Terms for Topic #. The top topics are represented by the light blue circles. Their centers are determined by the topic distances in relation to each other. The shorter the distance, the more similar the topics are. Speaking on the size of the circle, the larger the circle, the larger the topic distribution. The red in the Intertopic Distance Map represents what topic the user is currently viewing.

As the name suggests, the Top-30 Most Relevant Terms section shows the top 30 terms in a given topic circle. The red bars indicate the estimated term frequency within the selected

topic; how relevant the topic is, in relation to the other topics in the circle. The blue bars represent the overall term frequency. The λ slider shows the ranking of terms based on term relevance. The terms of a given topic is ranked in decreasing order (top to bottom) based on their topic-specific probability (λ = 1) (Sievert and Shirley, 2014). The λ slider involves the ranking of terms in regard to how relevant the terms are to the selected topic. A value of λ = 0 indicates the selection of terms that are most relevant to only the selected topic, and will rank terms higher if the term doesn't appear in any other topic.

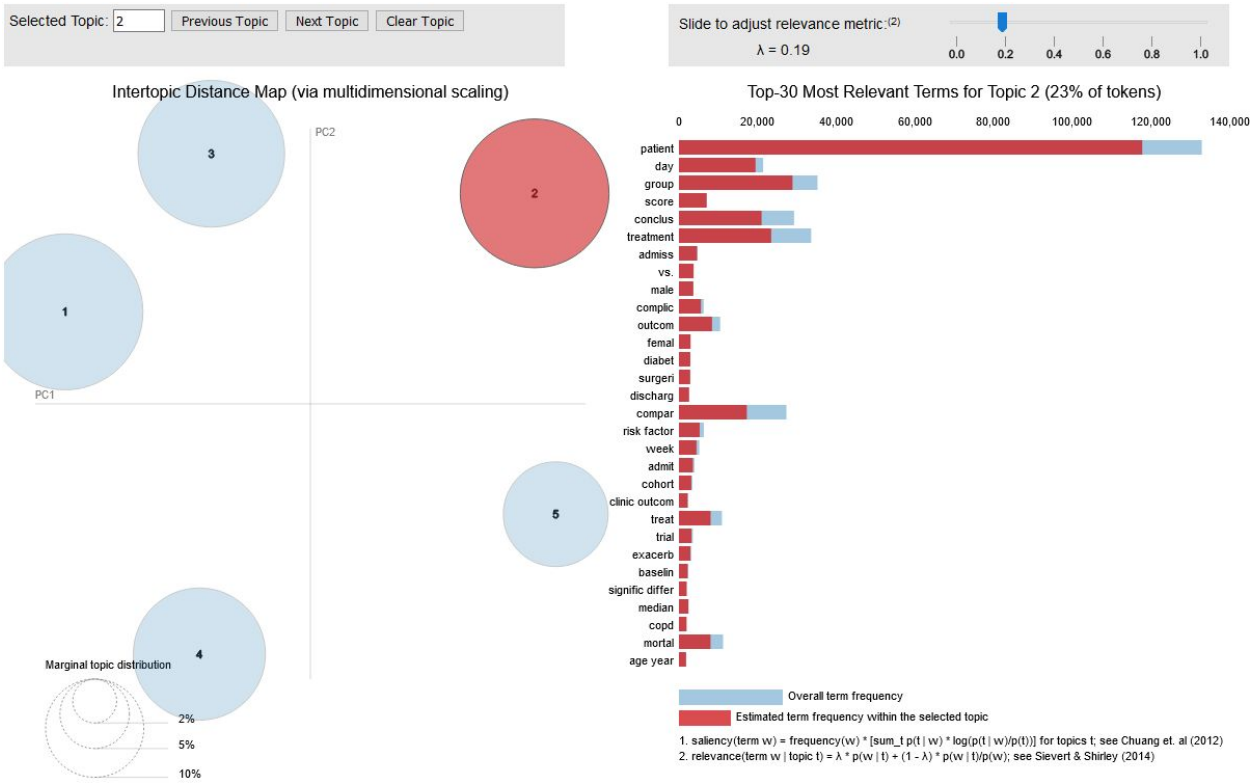

Figure [6]: The pyLDAvis visualization of the results of performing topic modeling with LDA.

3.1 Visualization techniques (word2vec)

Word2vec gives us an efficient word embedding for our corpus where each word is assigned a high-dimensional vector representation. The similarity of two word vectors can be determined by their cosine distance, meaning similar words or words that appear in similar contexts will be placed "close" to one another in a high-dimensional semantic vector space (Mikolov et al. 2013). This is an attractive property for machine learning models; however, this same property makes it hard for humans to gain an intuition of the meaning of word embeddings.

To visualize the high-dimensional word vectors, the t-SNE library from scikit-learn was used. The t-SNE algorithm's performance benefits from an initial dimensionality reduction if the initial number of features is large. Since the Word2vec model resulted in a high-dimensional embedding, we used PCA for the initial dimensionality reduction before applying t-SNE. The t-SNE library took parameters for tuning perplexity, the dimension of the embedded space, the random state the algorithm starts at, and an initial method for reducing the dimensions.

3.2 RESULTS AND ANALYSIS (LDA)

Overall, the results from LDA were expected. We saw health related discussions, research related language, and medical topics. Given that the research abstracts come from journals in the field of epidemiology, these results make sense. It was interesting to see the various ways in which researchers named the coronaviruses and the deadly SARS outbreak in 2002-2004, which was an ongoing research topic prior to the current pandemic, as well as the appearance of different names in different topics. However, there are places where the results are

less than ideal since the stemmer creates less than human-readable results and fails to sufficiently stem certain tokens (i.e. “viru” and “virus” appear in the same topic sometimes). This may explain why stemming resulted in a better average coherence and perplexity score, as incorrectly stemmed tokens would have the same semantic content, and likely would appear in the same topic. Also, there are some domain-specific stop words that were not removed during preprocessing, which were not caught due to our lack of domain knowledge. This was expected as our preprocessing steps were generic. Improvements could be made for future implementations.

The results of the abstract broken down by journal in comparison to the results of all abstracts are different due to the influence of COVID-19. This overshadows much discussion about previous research topics from other journals. This is expected since the composition of research articles from each journal will reflect the explosion of research being conducted to tackle the current pandemic. One journal in particular had interesting results that no other journal contained. In the *Surgical endoscopy* journal, there are topics related to surgical equipment rather than procedures, conditions, or symptoms for coronaviruses. The *Journal of virology*, *bioRxiv*, and *Virology* journals both contain substantially more technical topics than the *Surgical endoscopy* journal and the *PLoS One* journal. This can be accounted for as a result of the focus of each journal: three focused on biological and virological topics, one focused on surgical application and implications, and one that was more general about scientific topics. As a whole, the abstracts are much more heavily focused on the current coronavirus cases as those articles are the ones that have the greatest presence in the data set.

When comparing the results from all the abstracts with the year by year breakdown, we saw significant differences due to the over representation of COVID-19 related abstracts. The year by year breakdown showed an exponential increase in research going from only a few topics related to outbreaks, pandemics, and infection until about 2016. In 2018, there was an increase as concerns for an outbreak and infection, which further expanded until 2020, when every topic discussed coronaviruses and pandemic in some capacity. This was expected in some ways, but also unexpected in others. The explosion of research and discussion in 2018 and 2020 was expected since the pandemic is still ongoing. However, it was unexpected that the mention of outbreak was in articles prior and as early as 2016. In addition, there appears to be an outlier in 2010 where a rise of discussion of infectious diseases and pandemics likely as a result of the Swine Flu in 2009-2010.

All (107032 abstracts)	1: genom, applic, divers, dynam, ibv, evolut, cov, plant, platform 2: score, vs., male, femal, diabet, surgeri, discharg, clinic outcom 3: covid-19, pandem, care, recommend, coronaviru diseas, public health 4: patient, cell, express, inhibit, replic, sars-cov, mice, bind, vitro, cat 5: sampl, mers-cov, pedv, respiratori virus, bat, rsv, calv, pcr, diarrhea
PLoS One (1715 abstracts)	1: bind, regul, antigen, express, immun, induc, protein, antibodi, mice 2: estim, influenza, outbreak, risk, hospit, case, transmiss, mortal 3: detect, sampl, patient, sensit, assay, virus, test, clinic, pathogen 4: predict, structur, approach, inform, region, group, data, individu 5: strain, replic, viru, host, genom, anim, isol, infect, viral, inhibit, virus
Journal of virology (1600 abstracts)	1: sars-cov, respons, viral replic, treatment, pathogenesi 2: residu, mutat, mutant, domain, construct, conserv, structur, wild-typ 3: receptor, entri, function, glycoprotein, bind, mediat, virion, cell 4: mrna, rna, end, mhv, transcript, sythesi, gene, viral rna, genom 5: vaccin, anim, strain, isol, diseas, human, speci, virus, model
bioRxiv (1337 abstracts)	1: mutat, vaccin, sars-cov-2, covid-19, pandem, genom, design, structur 2: cell, ace2, inhibit, activ, express, drug, bind, treatment, sars-cov

	3: sampl, detect, pathogen, test, host, process, perform, virus, method 4: gene, patient, sequenc, rna, viral, novel, character, system, report 5: model, diseas, data, increas, provid, research, predict, studi, popul
Surgical endoscopy (887 abstracts)	1: oper time, report, year, without, procedur, instrument, min, techniqu 2: resect, case, success, approach, laparoscop, addit, feasibl, requir 3: vs., open, outcom, compar, differ, respect, rate, data, howev, surgeri 4: group, two gorup, day, min, time, studi, total, postop, compar, effect 5: measur, score, endoscop, assess, improv, surgeon, instrument
Virology (869 abstracts)	1: genom, rna, sequenc, encod, region, gener, function, structur 2: mice, infect, inhibit, cell, observ, dure, mous hepat viru, activ, detect 3: protein, express, mrna, virion, gene, membran, similar, fusion, encod 4: glycoprotein, mutant, site, structur, differ, membran, one, addit, virus 5: strain, recombin, coronaviru, result, mhv, two, mous hepat viru, bind

Figure [7]: Topics extracted from the corpus via LDA by top 5 journals. λ=0

2010 (3379 abstracts)	1: protein, cell, inhibit, mutat, replic, mice, bind, mediat, vivo, vitro 2: infecti diseas, research, practic, pandem, futur, problem, commun 3: surgeri, feasibl, procedur, complic, patient, oper, treat, safe, techniqu 4: estim, subject, valu, measure, concentr, higher, rate, particip, data 5: pneumonia, children, detect, influenza, isol, influenza viru, sensit
2012 (4268 abstracts)	1: score, versu, aim thi studi, week, postop, group, less, lower, day 2: cell, protein, express, inhibit, replic, receptor, pathway, mice, bind 3: health, public health, global, vaccin, infecti diseas, work, research 4: care, lesion, advanc, review, complic, recur, therapi, medic, remov 5: speci, sampl, sensit, detect, strain, influenza viru, isol, sequenc
2014 (5705 abstracts)	1: thi review, public health, infecti diseas, health, address, implement 2: procedur, complic, postop, recur, surgeri, train, laparoscop, surgeon 3: protein, cell, replic, bind, mutat, inhibitor, immun respons, recombin 4: control group, baselin, stroke, ratio, higher, lesion, score, vs., calcul 5: respiratori virus, pcr, pneumonia, children, influenza, detect, sampl
2016 (6298 abstracts)	1: trial, ventil, score, stroke, baselin, volum, average, measur, paramet 2: infecti diseas, commun, public, public health, issu, epidem, outbreak 3: aneurysm, complic, recur, control group, specimen, resect, underw 4: cell, inhibit, viral infect, immun, replic, mice vivo, pathogenesi 5: sequenc, genotyp, pcr, detect, sampl, speci, strain, assay

2018 (6216 abstracts)	1: cell, protein, express, gene, sequenc, inhibit, replic, antibodi, bind 2: trial, cohort, pain, clinic outcom, min, month, hospit, infant, pneumonia 3: infecti diseas, epidem, outbreak, surveil, nation, world, china 4: surgeri, lesion, surgic, postop, resect, procedur, devic, size, remov 5: qualiti, articl, systemat review, publish, meta-analysi, literatur, guidelin
2020 (32742 abstracts)	1: covid-19 pandem, practic, crisi, address, staff, servic, question 2: diabet, fever, children, adult, pediatr, standard care, admiss, male 3: china, transmiss, januari, wuhan, asymptomat, contact, coronavirus diseas 4: model, novel coronaviru, itali, forecast, simul, dyna, quantifi, dataset 5: sars-cov-2, cell, virus, protein, sars-cov, influenza viru, genom,inhibit

Figure [8]: Topics extracted from the corpus via LDA from 2010-2020. λ=0

3.2.1 Results and Analysis (Word Embeddings)

We used Word2vec to train two models with each of them having a different preprocess procedure. The corpus is the same for both models as well as the LDA corpus. We tested with changing the minimum frequency of word appearance so that we can see the similarity and cluster around the words that related to the topic of "COVID-19". Since the corpus was made from the many abstracts of many articles about the coronavirus pandemic, it is logical to assume that words related to the pandemic will have a high frequency. Due to this, we decide to increase the minimum frequency by stages to properly see the similarity between the common words that relate to the pandemic. With the model that has stop words removed, we started with a small minimum frequency and increased it gradually to "filter" out the words that don't get used that often from our corpus. The result is a much more clear and readable plot as the minimum frequency increases. As for the model without stop words removal, we started out with a high minimum frequency because the frequency of stop words will be pretty high so it's best that we

filter out as much as we could. We also compared the list of similar words for the word “patient” to see if the preprocessing method affected the model greatly or not. If the differences are noticeable, then it is safe to assume that the list of similar words for any word between the two models will be different. This will lead us to be able to identify what is the best way to preprocess with Word2vec.

Word2vec with stopwords removed:

Min word frequency = 100

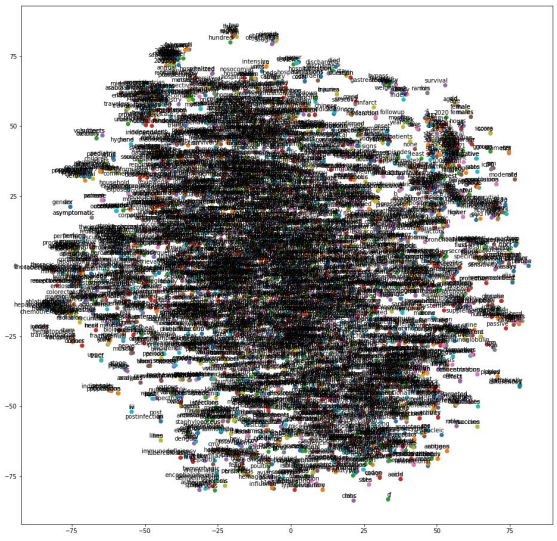

Figure [9]: t-SNE visualization when min_count=100.

Minimum frequency of each word = 15000

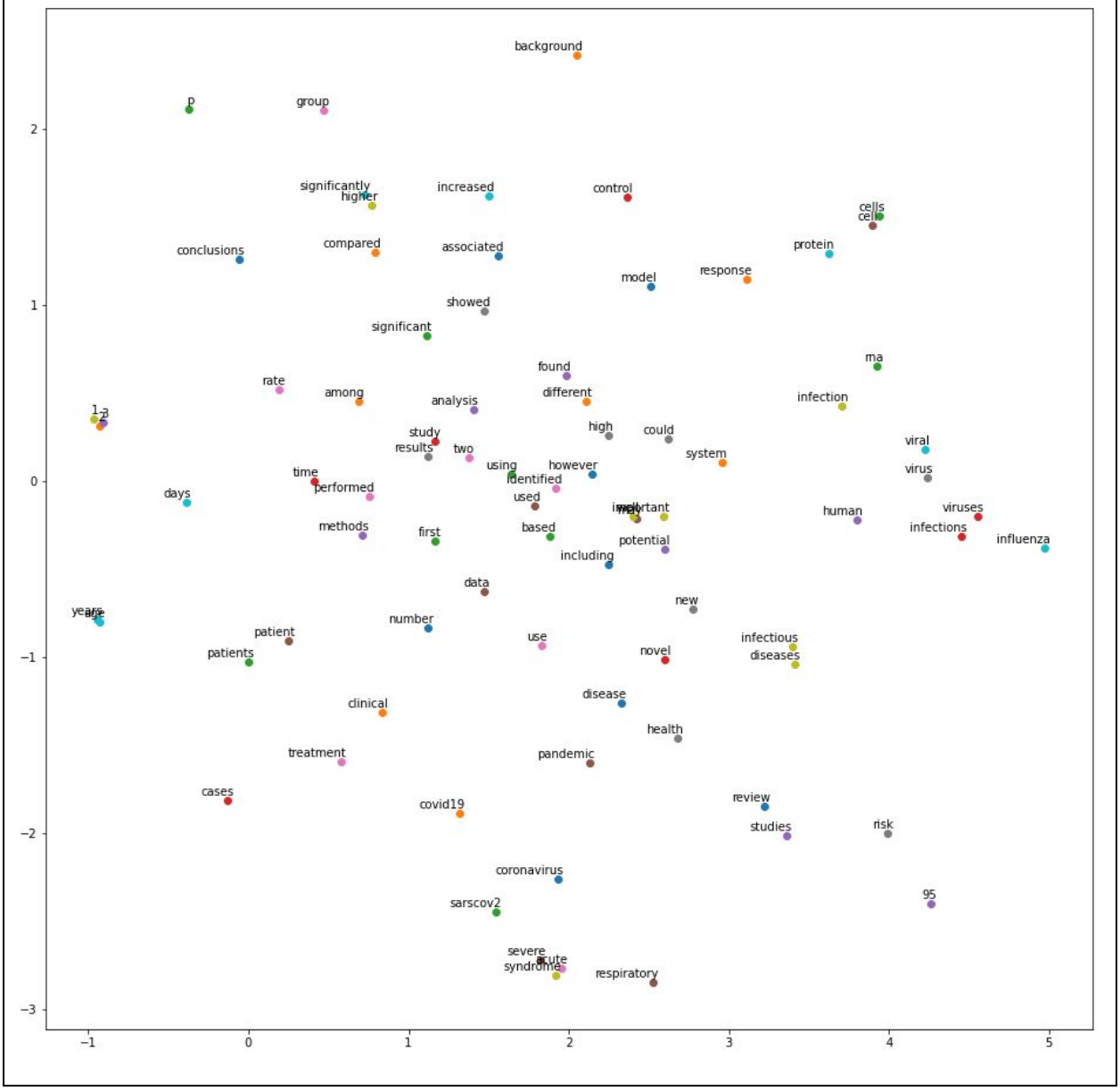

Figure [10]: t-SNE visualization when min_count=15000.

Minimum frequency of each word = 20000:

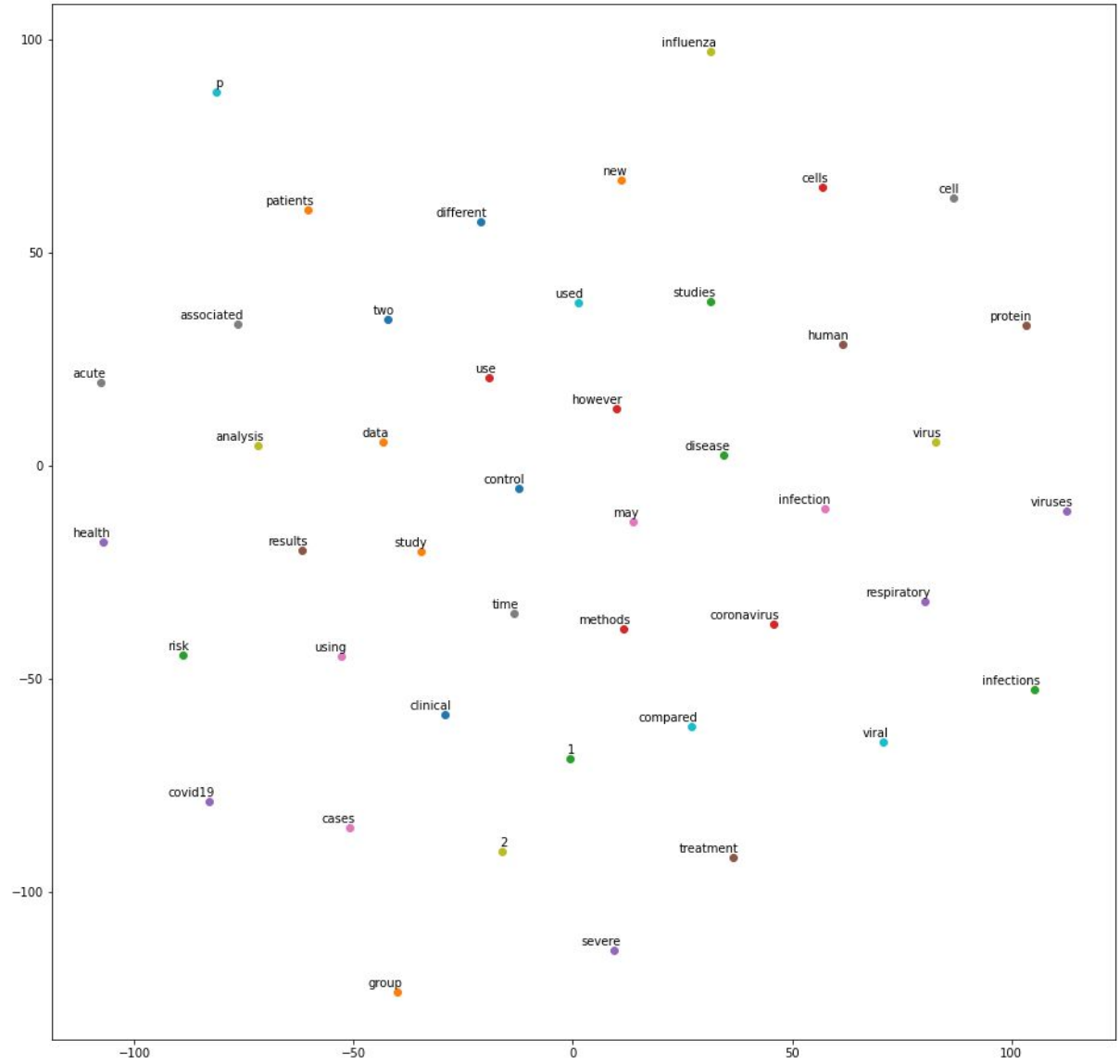

Figure [11]: t-SNE visualization when min_count=20000.

Trained on full corpus:
Given ‘patient’, the most similar words are:

Word	Similarity
woman	0.6127897500991821
patients	0.5534685850143433
case	0.5442690849304199
discharged	0.5367958545684814)
physician	0.4858970046043396
team	0.4787141978740692
managed	0.467002809047699
resolved	0.4546450972557068
ward	0.44436943531036377
discharge	0.43582770228385925

Figure [12]: words most similar to ‘patient’ with stop words removed.

Word2vec without stop word removal:
On full corpus, the most similar words to the word ‘patient’ are:

Word	Similarity
she	0.6319663524627686
her	0.6304579973220825
woman	0.6211405992507935
patients	0.5958157181739807
girl	0.5916122198104858

case	0.579210638999939
his	'0.5747033953666687
boy	0.5212789177894592
team	0.504177451133728
man	0.49293482303619385

Figure [13]: words most similar to 'patient' without stop words removed.

Minimum frequency of each word = 15000:

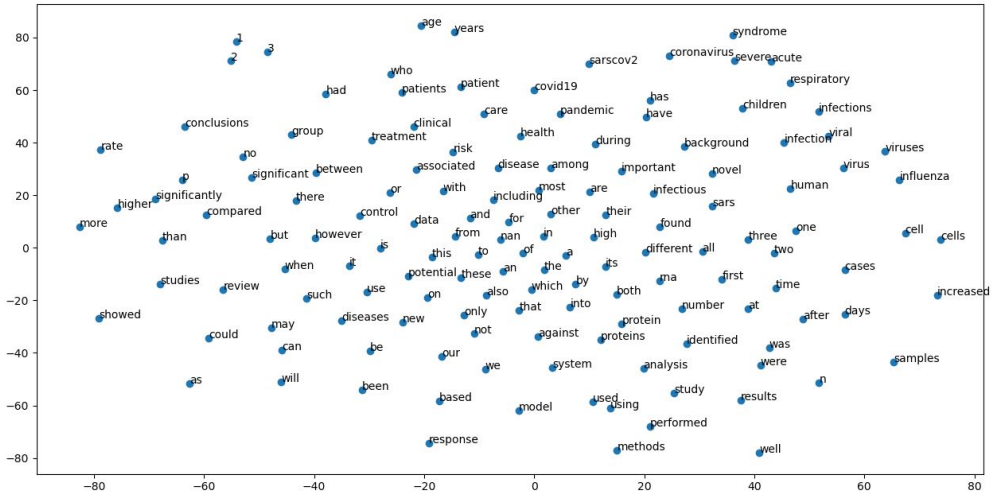

Figure [14]: t-SNE visualization when min_count=15000 (with stop words).

Minimum frequency of each word = 20000:

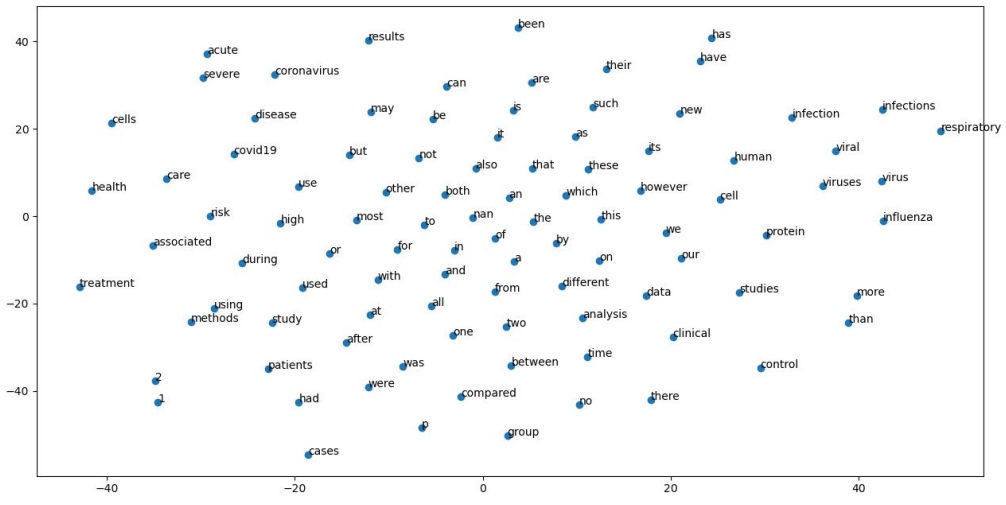

Figure [15]: t-SNE visualization when min_count=20000 (with stop words).

Based on the plots of the two methods of preprocessing for Word2vec, there are several key points to note. The difference between the amount of plot points is visible between the two methods. Without the removal of stop words, the plot is a lot more convoluted. For both methods of preprocessing our corpus, the resulting models showed words that are closely related to each other usually clustered together. For example, the words “coronavirus” and “infection” are plotted near each other because they appear together frequently in our corpus. This is a good representation of how Word2vec works. In addition, removing stop words does have an effect on the model. Some of the words similar to “patient” from the model that had stop words removed are words that depict more of “what happened” to the patient rather than “who” that patient is (i.e

pronouns) like the model that included stop words. Furthermore, the level of similarity of the word “patient” is also different. Notice that the word “woman” existed in both of the lists of similar words, but the degree of similarity is different. This is another reason why we think removing the stop words may have an effect on the resulting model.

4 DISCUSSION

LDA was made for topic modeling so using the gensim library for this task was relatively straightforward. The knowledge required to interpret the visualizations and information were more nuanced but manageable. The most significant task was understanding the roles and techniques for preprocessing since they are the most impactful aspect of the pipeline to the final results for the LDA methods we utilized.

Although Word2vec is not designed for topic modeling, it does make a very good visualizer for the topics that were discovered by the LDA algorithm. The gensim library was straightforward and simple enough for us to use without much hassle and the resulting vectors do indeed show some of the topics from LDA. From the plots above in section 3.2.1, we can see that words like ‘virus’ and ‘infection’ are closer to each other than ‘virus’ and ‘treatment’ which make sense given the context of the still ongoing pandemic in 2020. It also makes sense when we increase the frequency of words from low to high that the words related to the coronavirus remain to be plotted as more and more abstracts are mentioning the virus.

5 Limitations

During the course of this project, there were a number of conflicts which limited or blocked our initial research direction. The first obstacle was in obtaining access to the dataset. The initial dataset was to be tweets related to the COVID-19 epidemic, which included geographical coordinates and Twitter metadata about the language of the tweet. However, access to tweets requires explicit developer approval from Twitter, for which the researchers of this project continue to wait. This obstacle required choosing a different dataset, however, the chosen dataset contained text in various languages without a column for identifying the language of the text. The preprocessing step should filter out any tokens that do not have at least one character from the basic ISO latin alphabet, which should remove any non-latin based scripts.

The original aim of this project was to compare the performance of conducting topic modelling with LDA and word2vec. However, topic modeling is not possible with only word embeddings. Topic clustering, on the other hand, is more commonly associated with Word2vec and clustering algorithms, such as K-means clustering. In topic modelling, LDA and its predecessors, Latent Semantic Indexing/Analysis (LSI/LSA) and probabilistic LSI (pLSI/pLSA), learn some number of latent topic distributions in a corpus. In topic clustering, word distributions are partitioned into some number of clusters. While these two areas of NLP are similar, they are still inherently different, which did not allow for a direct comparison of the two.

6 Future Work

Future work for this project includes addressing the limitations mentioned above, expanding into either topic modelling or topic clustering methods, and using trained word vectors via LDA and Word2vec to complete a different NLP task that *can* be used to directly compare the performance of the two models. To address the presence of various languages in the text, a character-level language identification model may be used to filter out non-English abstracts.

In the direction of topic modelling, the usage of distance metrics exposed by the gensim library may be used to quantitatively compare topic distributions. This includes the Hellinger, Kullback-Leibler (KLD), and Jaccard similarity metrics. These metrics can be used to quantify the similarity of topics by journal or by years. Other than this, the performance of various topic modelling methods, such as LSA, pLSA, and random projections. As for topic clustering, there are a variety of models to generate word embeddings, a selection of clustering algorithms to generate topic clusters, and a number of dimensionality reduction algorithms to visualize the clusters. However, the original intent of this research is to focus on topic modelling.

This research project narrowed the scope of the corpus to only the abstracts of research papers in the medical field, specifically epidemiology, as relates to analyzing the COVID19 pandemic. Future work may expand upon this by using full research papers rather, or by including other fields of medicine. The researchers of this project note that using full research papers for topic modelling will introduce a significant hit to runtime performance and LDA

performance without the usage of vocabulary reduction and harsh keyphrase candidate identification techniques.

7 References

Blei, D. M., Ng, A., & Jordan, M. (2003). *Latent dirichlet allocation*. Journal of machine learning research (3).

Cambridge University Press. (2008). *Stemming and Lemmatization*. Stemming and lemmatization. https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

COVID-19 Open Research Dataset Challenge (CORD-19). Kaggle. (2020, June 10). https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge?select=metadata.csv.

Gensim: Topic Modelling for Humans. Radim Rehurek: Machine Learning Counseling. (2019). https://radimrehurek.com/gensim/models/word2vec.html.

Goldberg, Yoav, Levy, & Omer. (2014, February 15). *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. arXiv.org. https://arxiv.org/abs/1402.3722.

greenunknown/cs445-545-ml-group-project. GitHub. (2020). https://github.com/greenunknown/cs445-545-ml-group-project/blob/master/src/lda.py.

Kapadia, S. (2019, August 19). *Evaluate Topic Models: Latent Dirichlet Allocation (LDA)*. Medium.

https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0.

Kumar, K. (2018, May 3). *Evaluation of Topic Modeling: Topic Coherence*. DataScience+. https://datascienceplus.com/evaluation-of-topic-modeling-topic-coherence/.

Lee, H., Kihm, J., Choo, J., & Stasko, J. (2012, June 25). *iVisClustering: An Interactive Visual Document Clustering via Topic Modeling*. Wiley Online Library. https://onlinelibrary.wiley.com/doi/full/10.1111/j.1467-8659.2012.03108.x?casa_token=8ziHdV1SfhEAAAAA%3Awx77BurbZvDZRYDiAJOUs_sJvhaHz3WGsTYGTInQWS_pYI-TZp2DWS6OcGK9nVNlpWXGwFSgIx3Y0tIW.

Maaten, L., Hinton, G. (2008, November). *Visualizing Data using t-SNE*. Journal of Machine Learning Research. 9. 2579-2605.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781.

Nikita, N. (2016, February 1). *A tale about LDA2vec: when LDA meets word2vec*. Data Science Central. https://www.datasciencecentral.com/profiles/blogs/a-tale-about-lda2vec-when-lda-meets-word2vec.

Perone, C. S., Silveira, R., & Paula, T. S. (2018). Evaluation of sentence embeddings in

downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Pleplé, Q. (2013, May). Perplexity To Evaluate Topic Models. http://qpleple.com/perplexity-to-evaluate-topic-models/.

Pleplé, Q. (2013, May). Topic Coherence To Evaluate Topic Models. http://qpleple.com/topic-coherence-to-evaluate-topic-models/.

Prabhakaran, S. (2020, April 16). *Topic Modeling in Python with Gensim*. Machine Learning Plus. https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/.

Rong, & Xin. (2016, June 5). *word2vec Parameter Learning Explained*. arXiv.org. https://arxiv.org/abs/1411.2738.

Savoy, Grefenstette, L., T., Griffiths, B., J., Tenenbaum, … Lafferty. (1970, January 1). *Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora*. Information Retrieval Journal. https://link.springer.com/article/10.1007/s10791-012-9200-5.

Sievert, C., & Shirley, K. (2014, June). LDAvis: A method for visualizing and interpreting topics. *In Proceedings of the workshop on interactive language learning, visualization, and interfaces* (pp. 63-70).

Singh, D., Salakhutdinov, & Ruslan. (2018, January 5). *Knowledge-based Word Sense Disambiguation using Topic Models*. arXiv.org. https://arxiv.org/abs/1801.01900.

Stevens, K., Kegelmeyer, P., Andrzejewski, D., & Buttler, D. (2012, July). Exploring topic coherence over many models and many topics. *In Proceedings of the 2012 Joint*

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 952-961). Association for Computational Linguistics.

Wei, X., & Croft, W. B. (2006, August). *LDA-based document models for ad-hoc retrieval*. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 178-185).

Wieringa (2018, July). *Using pyLDAvis with Mallet*. From Data to Scholarship. https://jeriwieringa.com/2018/07/17/pyLDAviz-and-Mallet/