Exploring Topic Models for CORD-19: COVID-19 Academic Papers

Khachatur Mirijanyan
Udacity - School of AI
Machine Learning NanoDegree
khachatur.mirijanyan@gmail.com
May 1, 2020

Abstract

In recent times, the novel corona virus has spread throughout the globe and put the world under lock down. In response, medical professionals have begun looking for solutions to this worldwide pandemic through all potential means, new and old. However, progress has been slow due to the lack of knowledge about the virus. Currently, there is an abundance of information in the form of academic papers for all topics, including those related to COVID-19, but being able to filter through this information quickly has been difficult. A multitude of organizations have come together to release CORD-19, an easy to parse data set containing thousands of research papers that are potentially connected to COVID-19.

The goal of this project was to explore the contents of the data set and establish similarities and connections between academic papers, as well as what kinds of topics are present in CORD-19. Various topic modeling techniques were used to establish the number of topics within the data set, as well as how coherent these topics were. The results of the project was an increase over the benchmark topic coherence score to .663, with a relative standard deviation of the topic coherence of 0.144.

Index Terms - Topic Modeling, COVID-19, CORD-19, NMF, LDA, NLP, AI

1 Introduction

1.1 Problem Statement

There is currently a great collection of academic literature available to researchers and medical professionals to help understand and combat against the new COVID-19 pandemic. But even with the manpower available, it can be difficult to determine what papers are worth reading, what to expect in the paper at a glace, or if a paper is relevant to another one. While documents can be skimmed or abstracts can be read, that may not be enough information for researchers to determine whether or not spending to spend the time analyzing a certain document or if it is relevant enough to their own research.

Topic Modeling can approach this problem in a multitude of ways. First, the technique can utilize the word distributions and their relative proximity to one another to generate a number of topics comprised of important words. The model can then be used to generate a topic distribution for a document and determine which topics the document is comprised of and how strongly associated with a certain topic a document is.

1.2 Project Overview

The project will involve using the CORD-19 data set to create a topic model. CORD-19 is a publicly available set of research documents compiled by the likes of the US Government, Google, and Facebook that could potentially help in better understanding COVID-19. The goal of the project is to use the text in the research documents to generate a topic model with a suitable average topic coherence with a low relative standard deviation of topic coherence scores.

Since the project has not been done before, a benchmark using a simple LDA had to first be created. The data also had to be pre-processed and prepared for model training. Then, various smaller LDA models using only the paper abstracts and default model behavior were created and analyzed to obtain a better understanding of the number of topics as well as their computation times. The project was then slightly expanded to also include NMF models to see if a different type of model could achieve better results.

Lastly, the full text of the documents was used to create a set of models. This was an iterative process of creating a large model, and tuning the text processing until the words in the topics of the model improved. A set of models with varying topics was then made and the one with the best scores was chosen. All models were created using the Gensim python package.

1.3 Metrics

For evaluation, two quantitative metrics were used, as well as a qualitative evaluation. The first quantitative metric was the average topic coherence score of each topic, specifically using c_v topic coherence.

A topic coherence value estimates how coherent the

extracted words of a topic are. For example, the top 20 words of an identified topic can be used to determine how coherent a topic is. A further explanation of using coherence values in many different models can be found in *Exploring Topic Coherence Over Many Models and Many Topics, Stevens et. al* [1].

The coherence value I am using in particular is c_v, which is shown to have the best performance based on *Exploring the Space of Topic Coherence Measures, Roder et. al*[2]. Through these papers we can also see that there is a connection between effective models and good coherence measures. In the case of c_v, the measure is between 0 and 1, with 0 being non-coherent and 1 being overly coherent. The topic coherent value will determine how good a model is. An average value around 0.7 mean that the topic is coherent, but it is not filled with words that are either overly related or are too similar. A very high topic coherence could signify too many topics or over fitting.

Relative standard deviation of topic coherence looks at the spread of topic coherence values across all topics. If this number is high, that means there are topics with wither too high or too low coherence, which could be a sign of too many topics. The qualitative evaluation is simply done by looking at the words to see if the topics make sense. An example of this was when a topic had what seemed to be acceptable values, but consisted almost entirely of Spanish and French stop words.

2 Analysis

2.1 Data Exploration

The machine learning field that the solution is being designed for is under that of natural language processing and natural language understanding. For this project, the CORD-19 (version 8) was chosen. The data set was created by the US Government and large technology companies, and distributed through Kaggle in an easy to parse format.

The use fullness of the topics models is largely dependent on the text input into the model. The most important factors for a good topic model are the number of unique tokens/words, the types tokens, and the number of documents. When conducting exploration and making exploratory models, only the titles and the abstracts were used. The last sets of models generated use the full extracted text.

The data set also comes with a metadata file with a lot of useful information such as which files have the full text. In total there are 51K documents in CORD-19, but there are discrepancies between the metadata and the actual files provided, such as files that do not exist, or duplicates of files. Many files do not have abstracts or titles in the metadata and thus not in the full text provided either. After resolving the discrepancies between the metadata information and the actual files and taking only documents that contained the full text and those with an abstract, there remained 33,500 documents.

2.2 Benchmark

Since no topic model of CORD-19 data exists, a simple model had to be created as a reference point. This required simple pre-processing steps as well as using an out of the box version of Gensim's LDA implementation. The benchmark also only used the abstracts and titles of the text. The pre-processing at the benchmark phase was also not as refined as later in the process.

The text was processed and a dictionary was created that filtered out words that occur in less than 20 documents and more than 50% of documents. Then a bag of words corpus was created from that dictionary. The resulting corpus contained nearly 11,000 unique tokens.

The model used nearly all the default values present in Gensim's LDA implementation. This meant an assumed number of 100 topics. The only change made to the hyper parameters was the number of passes through the corpus in training, which was changed to ten from the default one.

Using a c_v coherence model, the top word were used to represent a topic. The average topic coherence and relative standard deviation of the topic coherence were .535 and .281 respectively. This is not very good. The average topic coherence for a working model would need to be at least .6. The relative standard deviation value is what truly makes the benchmark unsatisfactory. The value indicates that there is a standard deviation from .535 of 30%. This would result in many topics having coherence score very close to 1 or very close to 0, making those topics be unsatisfactory representations of text.

2.3 Exploratory Visualization

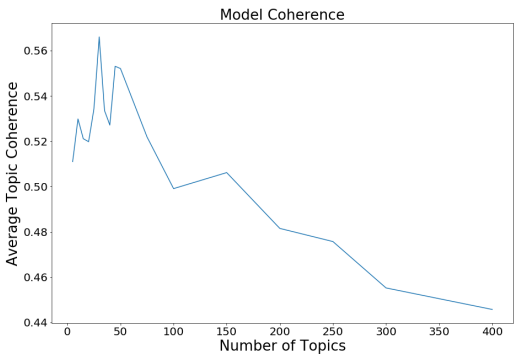

Figure 1: There is an increase in the topic coherence scores originally, and then drastically decreases

Before using the full text, a suitable range for the number of topics had to be found. The default of 100 topics may have been what was worsening the metrics. To discover this optimal range, as well as understand the computational limitations of this project, a series of models were generated using the same data as in the benchmark. The only difference between the models is the number of topics selected.

As seen in Figure 1, the result of this exploration re-

sulted in the topic coherence score fluctuating wildly with the number of topics. The optimal range for the number of topics looks to be about 30 to 40 for the abstracts. This does not necessarily indicate that this would be the optimal range for the full text. The full text will have more unique tokens and thus could have more topics that need representing.

This exploration was used to determine that the full text models should have topics between 10 and 100. Also, the computation times showed that while the first few models with low numbers of topics could take about 2-3 minutes, the model with 400 topics takes about 30 minutes. This would get further increased when creating the full text and was part of the considerations when deciding the topic range used for the full text data set. Also, the benchmark topic coherence value and the value seen in the chart for 100 topics are different. This is due to multiple runs of the benchmark and random initialization for the model. This is why the passes are set to 10. A better model with more refined text processing and a more accurate selection for the number of models will vary less over multiple runs.

2.4 Algorithms and Techniques

Below, the primary algorithms and techniques are discussed. They are Latent Dirichlet Allocation (LDA) and Non-Negative Matrix Factorization (NMF). LDA was used at all steps of project, while NMF was only used very late during the exploration step as an addition and during the full text steps. The text representation used was Bag of Words (BoW)

2.4.1 Bag of Words

The bag of words technique of words representation is effectively a set of unique tokens. The data is first processed and transformed into a dictionaru with the counts represented in the words. The would are filtered for extreme occurrences of words and then turned into a corpus as a bag of words. The LDA model then uses the bag of words as a word bank and can be further supplemented with the dictionary count.

2.4.2 Latent Dirichlet Allocation

The original focus of this project was to use LDA for topic modeling. LDA is a generative statistical model and in the context of NLP, attempts to find the latent unobserved semantic relationships between documents and words. The model then uses these identifies these word relationships to generate topics comprised of words or tokens. The first major use of LDA in text modeling is described in *Latent Dirichlet Allocation, Blei et. al* [3]. LDA is specifically a Beyesian model which "each item of a collection is modeled as a finite mixture over the underlying set of topics". LDA is the backbone of many modern topic models.

2.4.3 Non-Negative Matrix Factorization

The use of NMF was added later on in the project to diversify the types of models used. It is a linear algebra technique for factorizing a matrix into two constituent parts with the property that all matrices have no negative elements. V is the matrix representing the data, while H and W are the parts.

$$\begin{bmatrix} V_{11} & V_{12} & V_{13} & . \\ V_{21} & V_{22} & V_{23} & . \\ V_{31} & V_{32} & V_{33} & . \\ . & . & . & . \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & . \\ W_{21} & W_{22} & . \\ W_{31} & W_{32} & . \\ . & . & . \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} & H_{13} & . \\ H_{21} & H_{22} & H_{23} & . \\ . & . & . & . \end{bmatrix} \quad (1)$$

H and W are generated and would approximately reconstruct back to V when multiplied. The technique was discussed in great length in *A Practical Algorithm For Topic Modeling With Provable Guarantees, Arora et. al* [4]. This results in a model that is faster and may have more accurate topics depending on the semantic variance of the documents. For this particular data set, it seems that NMF produced consistently better results.

3 Methodology

3.1 Data Pre-Processing

Extensive pre-processing had to be conducted at all stages of the project. The first major challenge was data selection. There were many inconsistencies existed between the metadata document and the actual files available. There were duplicates present in the metadata that had to be eliminated. There were files that existed in one place that did not exist in another. Furthermore, only files with abstracts and full texts could be selected. This resulted in a shrinking of the number of document from over 50K to just over 33K.

Next came the processing of the text. Obtaining the abstracts was easy as it was available in the metadata. Obtaining the text in the body of the paper was slightly harder, but the data was represented in an easy to parse .json format, so there was not much of an issue there. Processing the text involved lowering the case of all the words, tokenizing the text, and removing unwanted tokens. It was the 3rd part of the text processing that required the most work and refinement.

3.1.1 Text Refinement

The original text processing stage was very simple. It removed some English stop words, removed all numbers, and removed all tokens with length less than 2. Through multiple iterations of model construction, this was further expanded from the benchmark all the way to the final models. The set of stop words became more robust as stop words were pulled from the NLTK stop word collection. This was further enhanced by also including French and Spanish stop words. This was due to some topics being almost entirely comprised of foreign language stop words like "el" and "la".

The numeric token elimination was also refined. Since the documents are medical in nature, many tokens represented medical terms which were a combination of words and numbers such as "H1N1". To make

sure to keep these kinds of values, regular expressions were used to only remove lone numeric values as well as various kinds of punctuation and non alphanumeric characters.

At the end of the refinement process, a dictionary was created of the tokens with the extreme words filtered. A bag of words corpus then generated out of the dictionary. This resulted in around 11K tokens for the small text models and about 58K tokens for the full text models.

3.2 Implementation

For the models themselves, the hyper parameters passed through did not change much from the early models. The most important parameter for topic modeling is the number of topics, and like many other unsupervised clustering algorithms, the best way to determine the correct amount of clusters is to generate many models with different cluster amounts and compare them.

In both the LDA and NMF models, the number of passes through the corpus was increased to 10. An ID to token representation was created from the dictionary and also added during model training as it seemed to increase performance. Gensim's implementation of LDA also had a version that could utilize multiple cores and provided more efficient CPU usage, further speeding up model training. Although LDA multi-core improved LDA computation times, it was still slower than NMF. On the full text, both LDA and NMF models were generated on topic numbering 10, 15, 20, 25, 30, 35, 40, 45, 50, 75, and 100. The hypothesis was that the best models would fall somewhere around 30 to 50, but one of the larger models may also do well as the topics become more and more focused the more of them that there are.

The model metrics were created using Genim's coherence model. Using c_v as the coherence value, the top 20 words from each topic were generated. LDA's topic coherence generation only required the documents and the dictionary while NMF's also required the corpus. After all the topic coherence for each topic were calculated, the average was taken, and their relative standard deviations was also calculated.

4 Results

Through testing of both types of models at different amounts of topics, it seems NMF does better across the board. Looking at Figure 2 and Figure 3, the NMF model performs consistently better for average topic coherence and topic spread respectively. From examining the actual content of the topics, it seems the words chosen for the NMF model make more sense at a glace.

4.1 Quantitative Results

By training the model on a number of topic amounts, a few key points stand out. Firstly, the NMF model is better using these metrics. Also when looking at the figures above, both models seem to do well around the

	ATC	RSTD
lda_40	.648	.216
nmf_40	.633	.144
nmf_75	.669	.177

Table 1: Results of chosen models. ATC: Average Topic Coherence. RSTD: Relative Standard Deviation

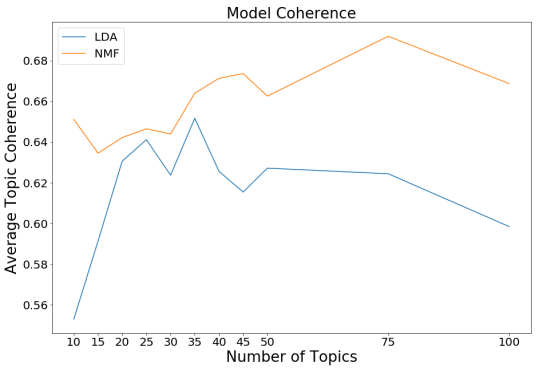

Figure 2: The topic coherence of the NMF models across all topic numbers seems consistently better

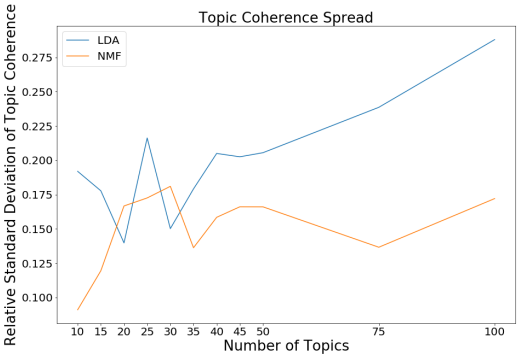

Figure 3: The spread values are similar until the number of topics increases. They then diverge

30-40 mark, the LDA model begins to perform much worse, while the NMF model improves a little bit or stays largely the same.

Of the three chosen models, nmf_40 performs best. It has nearly the same coherence score as nmf_75 and a much better score than lda_40. It also has the best spread which is very important. It is imporant to note that when looking at the individual topic scores, for both nmf_75 and lda_40, there were topics with coherence scores close to 1. These topics contain words that are too similar to one another. The largest topic coherence for nmf_40 was .83, which is still high and possibly over-fitting, but still not terrible. This increase was expected for nmf_75 since the increase in the number of topics would cause some of the topics to become useless. It is important to note that most of the models generated, not just the top 3 all performed much better than the benchmark. They had both better topic coherence and a smaller spread.

4.2 Qualitative Results

Judging by eye of the efficacy of the topics is difficult since I am not a medical professional. Still, from what I could see, the words in the topics did seem to make a lot of sense. For example here are the top 8 words from two of the topics in nmf_40.

Topic1 - *public, care, surveillance, medical, national, systems, services, countries*.
Topic1 - *cov, mers, sars, rbd, camels, covs, ace2, bat*.

It looks like the first topic is about government medical services and the second is about the different types of corona viruses and their relationship to animals.

5 Conclusions and Future Refinement

There are definitely some things that could have been done differently and future work that could improve the model. For text processing, an easy fix would be to lemmatize words and add measurement units into the list of stop words. Lematizing the words would remove pairs of words in the same topic such as ”protein” and ”proteins”. Removing unit measurements like ”kg” and ”cm” would help reduce needless information. A more complicated text processing task that could be done to improve the model is fuzzy matching similar words and representing them as a single word. This would reduce the amount of tokens and create a more generalized topic model.

For the model itself, the main focus would be on hyper parameter tuning. There are a lot of hyper parameters available for both the NMF and LDA models that were not used. For NMF in particular, it may be useful to use a TfIdf representation of the original text. NMF is is somewhat sensitive to outliers and a regularized text matrix could alleviate those issues.

Even with these issues, the project was a success. The model created performed much better than the benchmark on both average topic coherence and relative standard deviation of the topic coherence. Moreover, the number of topics selected seems to be a proper reflection of the topics in the text. While the values are adequate, the words in the topics themselves are clustered together in a way that makes sense.

References

[1] Keith Stevens, Philip Kegelmeyer, David Andrzjewski and David Butler. *Exploring Topic Coherence over many models and many topics*. Addison-Wesley, Reading, Massachusetts, 1993.

[2] Michael Roder, Andreas Both, and Alexander Hinneburg *Exploring the Space of Topic Coherence Measures*.

[3] David Blei, Andrew Ng, Michael Jordan. *Latent Dirichlet Allocation*.

[4] Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. *A Practical Algorithm For Topic Modeling With Provable Guarantees*.