Name: Vidhi Gupta (vg5vc)
Team: Haritha Guttikonda, Sania Rasheed

Due: 04/29/2020

DS5001 Exploratory Text Analytics

COVID-19 Research Data Analysis

1 Introduction

With the current novel coronavirus COVID-19 pandemic, a lot of research is going on to deal with this crisis. To aid the global research community, an open dataset of scholarly articles called the COVID-19 Open Research Dataset (CORD-19) has been prepared. The CORD-19 dataset contains around 57,000 scholarly articles, including over 45,000 with full text about COVID-19, SARS-CoV-2, and related coronaviruses. The goal is to derive new insights that can help in countering this infectious disease. Going over so many articles manually is tedious, hence in this project we have conducted an exploratory analysis of this dataset to get a better understanding of it.

2 Digital Analytical Edition of Corpus

2.1 Machine Learning Corpus Format (MLCF)

For the purpose of this project, we extracted the “abstract” and “result” sections of commercial and non-commercial published papers from their respective “.json” files. We used regex pattern matching to identify if the paper mentions COVID or coronavirus and tagged them as “covid_19” papers if they did. The “metadata.csv” file contains information about all the papers. Using this file we extracted the “paper_id”, “title”, “publish_year”, “publish_month”, and, “journal” for each paper and framed a consolidated “.csv” file that matches the MLCF Format.

Due to the restriction in terms of the computing resources available, I filtered papers that had been published since 2018 and found **2256** papers that consisted of both commercial and non-commercial papers.

2.2 Standard Text Analytic Data Model (STADM) and its NLP Annotation

The following three tables were generated to represent the papers such that they conform to the F2 model.

1. **LIB.csv**: This file contains all the metadata information about all the papers. This includes: paper_id, title, journal, publish_year, publish_month, type (commercial or non-commerical), and, is_covid19 (true if papers mention covid or coronavirus).

2. **TOKEN.csv**: The OCHO indexing used for tokenizing the papers is: [“paper_id”, “section_num”, “para_num”, “sent_num”, “token_num”]. The “section_num” indicates weather the token belongs to the abstract(marked 0) or result(marked 1). Tokenization was done using the *word_tokenize* method of the NLTK package. Tokens are lower-cased and punctuations and blank lines are removed to get terms. Each term is associated with a parts of speech (POS) tag using the *pos_tag* method of the NLTK package.

3. **VOCAB.csv**: The VOCAB table consists of all the unique terms that are present in the TOKEN table. Its count and other features that indicate if the token is a number, or a stopword. The stems, lemma and the POS tag most associated with the term is also present in this table.

2.3 STADM with Vector Space models

A TFIDF (term frequency inverse document frequency) representation is created for each paper using the sklearn implementation *TfidfVectorizer* and saved as **TFIDF.csv** with shared “term_id” with the VOCAB table. To ensure that the in-built implementation gives the same tokens, we override the in-built tokenization with our own tokenize module. Also, a “tfidf_sum” is calculated for each term and added to the VOCAB table.

2.4 STADM with analytical models

The annotated and vectorized model is extended using Principle Component Analysis (PCA), word2vec Word Embeddings, Sentiment Analysis and, Latent Dirichlet Allocation (LDA).

PCA:

PCA is performed using the sklearn implementation *PCA* on the TFIDF matrix generated earlier. Top 10 components are extracted, such that most of the variance in the TFIDF values is consolidated in these 10 dimensions. The PCA values are stored as the **LOADINGS.csv**.

word2vec Word Embedding:

The *word2vec* module of the gensim package is used to get 100 dimensional vectors to represent each word. Since our dataset consists of scientific journals not all words had a vector representation. The vector representation of the words is stored as **EMBEDDINGS.csv**. It contains the vector representation for each word along with 3 dimensional coordinates generated using t-SNE analysis.

Sentiment Analysis:

The sentiment associated with each term is computed using the *sentiwordnet* module of the NLTK package. This module gives the postive, negative and neutral polarity associated with the word. These polarity values are added to the VOCAB table and the updated table.

LDA:

As the dataset is large, two different topic models are generated, one for the abstract of all papers and the other one for the result of all papers. The *LdaModel* module of the gensim package is used to get the topic distribution per document which is stored as <**ABSTRACT or RESULT**>**_DOCTOPIC.csv** and the topic distribution per word which is stored as <**ABSTRACT or RESULT**>**_TERMTOPIC.csv**.

3 Inferences

Top 10 words based on TFIDF:

The top 10 words based on the tfidf values consist of key terms like: "patients", "viral", "infection". Given that we know the articles are on virus and infectious diseases these terms are expected. However, there are a few more terms like: "fig", "figure" and "p". The presence of these terms can be attributed to only taking the abstract and the result section of the paper. Almost every result section refers to figures and tables. It was interesting to see the word "p", this can be an indication that p-values are frequently used in clinical studies.

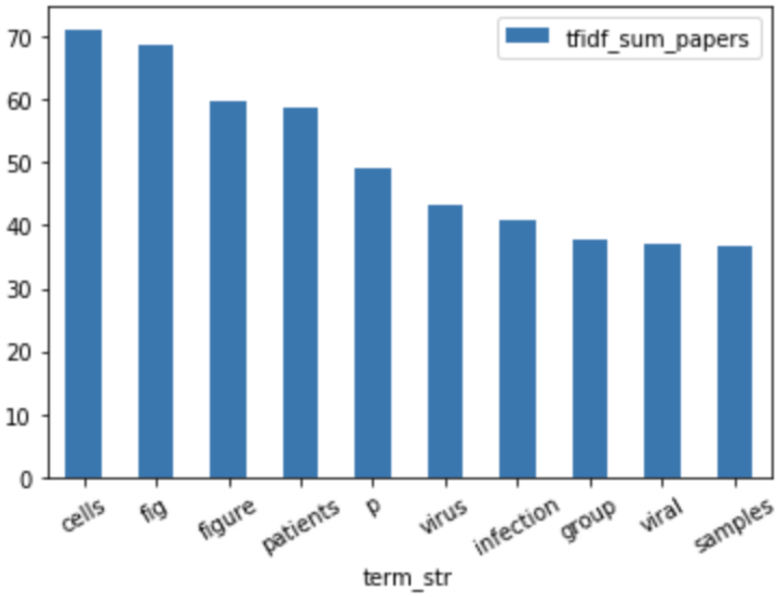
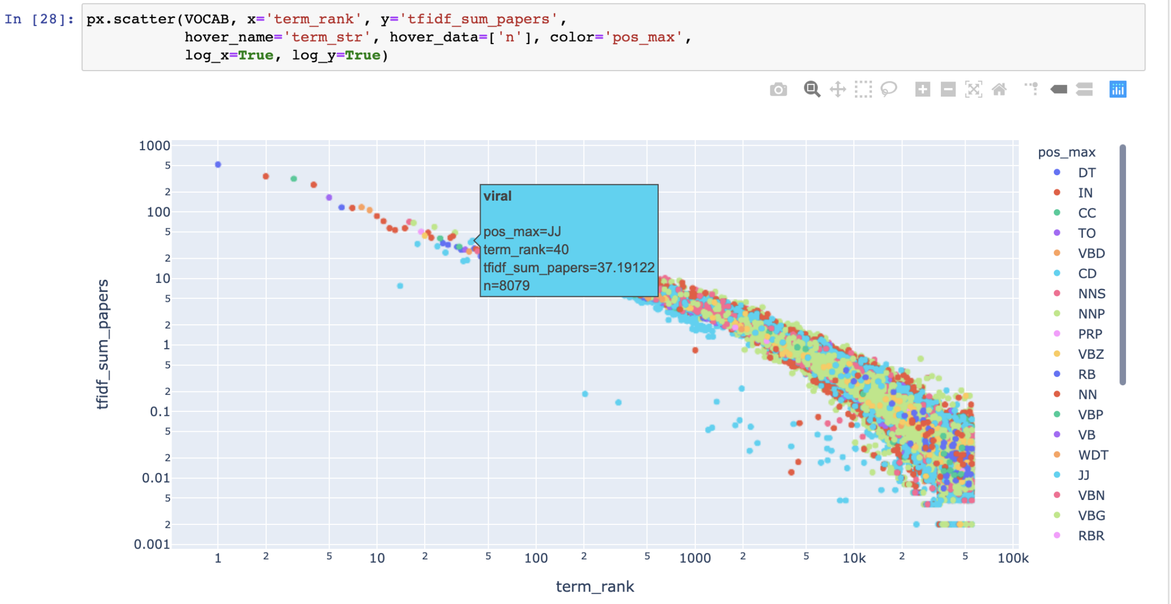


Figure 1: Top 10 words based on TFIDF sum

Term Rank vs. TFIDF:

I computed the term rank for each term based on the number of times they occur. On plotting the log of term-rank against the log of the tfidf-sum for each term it follows a decreasing pattern indicating that as the term count is decreasing so is the tfidf value. I expected the stopwords to be slightly off but that was not observed in the plot. This could be due to the large amount of data. While there are only 2256 papers, each paper contains lot of text. The term frequency could be too large such that it could not be normalized substantially based on the inverse document frequency.

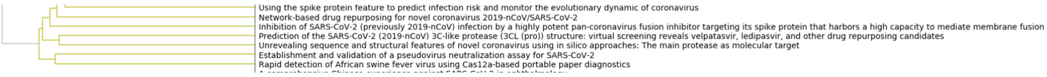


```
In [28]: px.scatter(VOCAB, x='term_rank', y='tfidf_sum_papers',
                    hover_name='term_str', hover_data=['n'], color='pos_max',
                    log_x=True, log_y=True)
```

Figure 2: log(term-rank) vs. log(tfidf-sum)

Clustering Papers of 2020 tagged as COVID-19:

I used hierarchical clustering on the subset of papers that were published in 2020 and tagged as COVID-19. There were 67 such papers. As mentioned above, papers that had a mention of coronavirus and covid were tagged as covid_19 papers. Even among these papers, some papers might not be very similar to the current virus we are fighting as viruses have the capability to mutate very easily. Hence I wanted to cluster documents and find which documents are the most similar to papers that are specifically on COVID-19. This can help researchers focus on specifically these papers. The clusters are created using the euclidean distance measure between papers. From 3 we can see that, the cluster of papers about the SARS-CoV-2 (virus that causes COVID-19) resembles a paper on the African Swine Flu and there might be something to explore and learn from.



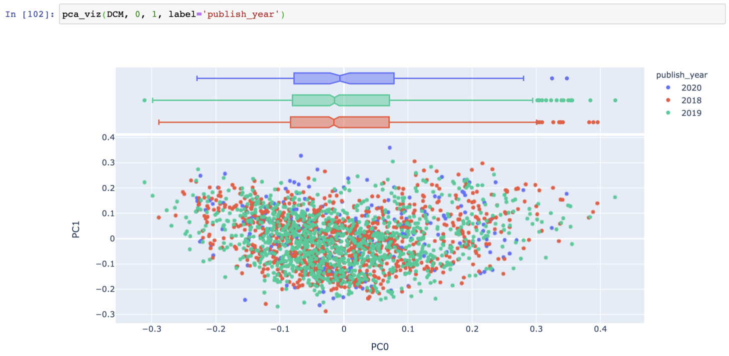
Using the spike protein feature to predict infection risk and monitor the evolutionary dynamic of coronavirus
Network-based drug repurposing for novel coronavirus 2019-nCoV/SARS-CoV-2
Inhibition of SARS-CoV-2 (previously 2019-nCoV) infection by a highly potent pan-coronavirus fusion inhibitor targeting its spike protein that harbors a high capacity to mediate membrane fusion
Prediction of the SARS-CoV-2 (2019-nCoV) 3C-like protease (3CL (pro)) structure: virtual screening reveals velpatasvir, ledipasvir, and other drug repurposing candidates
Unrevealing sequence and structural features of novel coronavirus using in silico approaches: The main protease as molecular target
Establishment and validation of a pseudovirus neutralization assay for SARS-CoV-2
Rapid detection of African swine fever virus using Cas12a-based portable paper diagnostics

Figure 3: Section of hierarchical clustering

Principal Component Analysis:

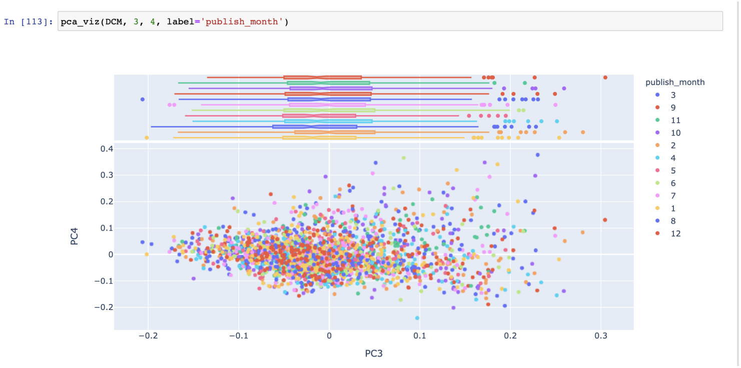
Using principal component analysis on the TFIDF representation, all papers are reduced to 10 dimensions using 10 principal components to incorporate most of the variance.

The PCA plot of PC0 against PC1 labeled by the publish year does not give much information. All papers are uniformly distributed. The same goes for the PCA plot of PC3 against PC4 labeled by the publish month. This plot also does not give us much information. In fact I tried various combinations of principal components against one another. However, as you can see in the figures, the papers are very uniformly distributed when it comes to time.



```
In [102]: pca_viz(DCM, 0, 1, label='publish_year')
```

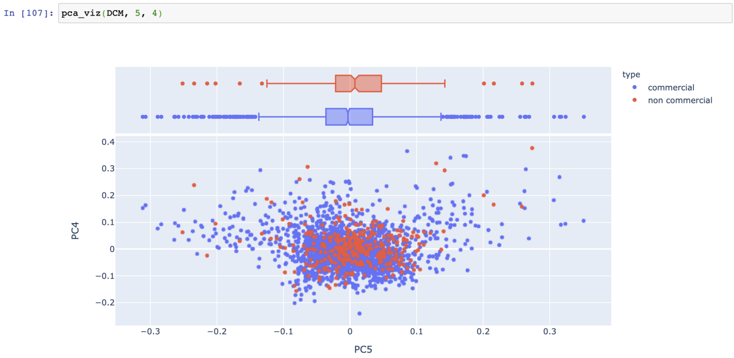
Figure 4: PCA labeled by publish-year



```
In [113]: pca_viz(DCM, 3, 4, label='publish_month')
```

Figure 5: PCA labeled by publish-month

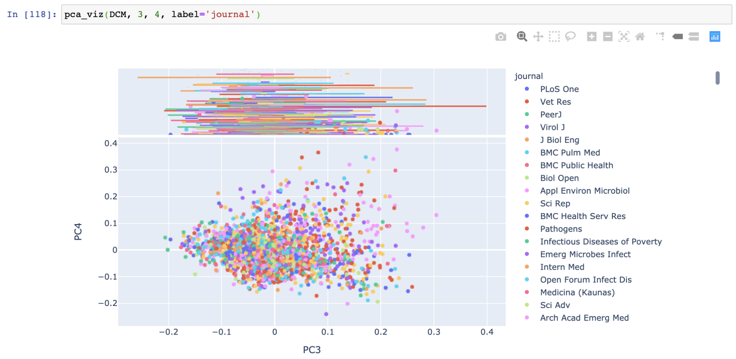
The PCA plot of PC5 against PC4 labeled by the type of paper (commercial vs. non-commercial) shows that while the median for both the categories is close by, the commercial type has more outliers and are more distributed across the fifth principal component compared to the non-commercial type.



```
In [107]: pca_viz(DCM, 5, 4)
```

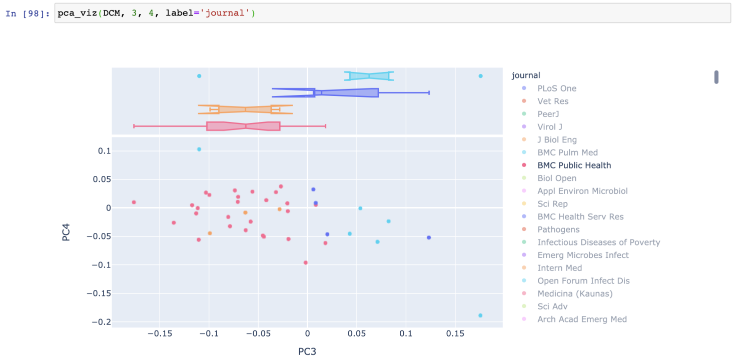
Figure 6: PCA labeled by type

The PCA plot of PC3 against PC4 labeled by the journal of the paper was not very useful when viewing all journals together, as there are 444 journals. However, if we select specific journals we can observe clusters. I selected 4 journals: “BMC Public Health” (pink), “Health Res Policy Syst” (orange), “BMC Microbial” (purple) and “J Clin Microbial” (blue). From the figure we can see a distinct separation between microbial and public health policy journals.



```
In [118]: pca_viz(DCM, 3, 4, label='journal')
```

Figure 7: PCA labeled by journal



```
In [98]: pca_viz(DCM, 3, 4, label='journal')
```

Figure 8: PCA labeled by journal (selected journals)

Sentiment Analysis:

To get the sentiment polarity associated with each term, we use SentiWordNet which is a lexical resource for opinion mining. It assigns each word three polarity values: positive, negative and objective (neutral) such that they sum up to 1. Most of the words in our dataset are medical words and do not have a specific polarity associated with it. These words are assigned a neutral polarity of 1. I use a univariate KDE plot for each of these scores, to observe the probability distribution. As we can see, the for both positive and negative polarities the density concentration is higher towards the lower values, whereas in case of the neutral polarity the density is way higher towards 1.

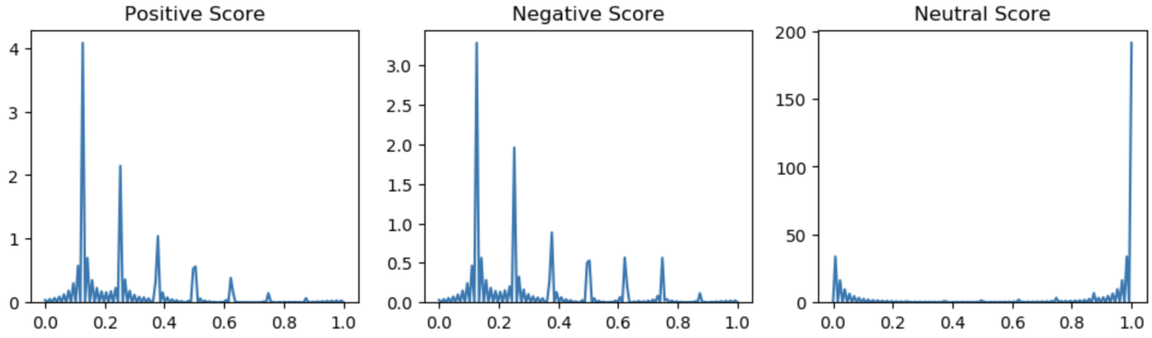


Figure 9: KDE Plot for Sentiment Polarities

Word Embedding t-SNE plot:

The word embeddings derived from the word2vec model are plotted in 3 dimensions using t-SNE. There were a total of 14916 words. Plotting them together made deciphering clusters almost impossible. Hence, I sampled 100 words and plotted them. Here, we can see that many biological terms like “ribosomes”, “enteroendocrine” and “chromatin” are clustered together. Another cluster of “magnitude”, “peak”, “fluctuation” and a few other words can be seen. The point to note here is that these are pre-trained word embeddings and not really derived from the text we have. Hence they represent the generalized meaning of words and not in the context in which they would have been used in the papers. Training a word2vec model from scratch will be able to better model these words based on the context in which they have been used in the papers.

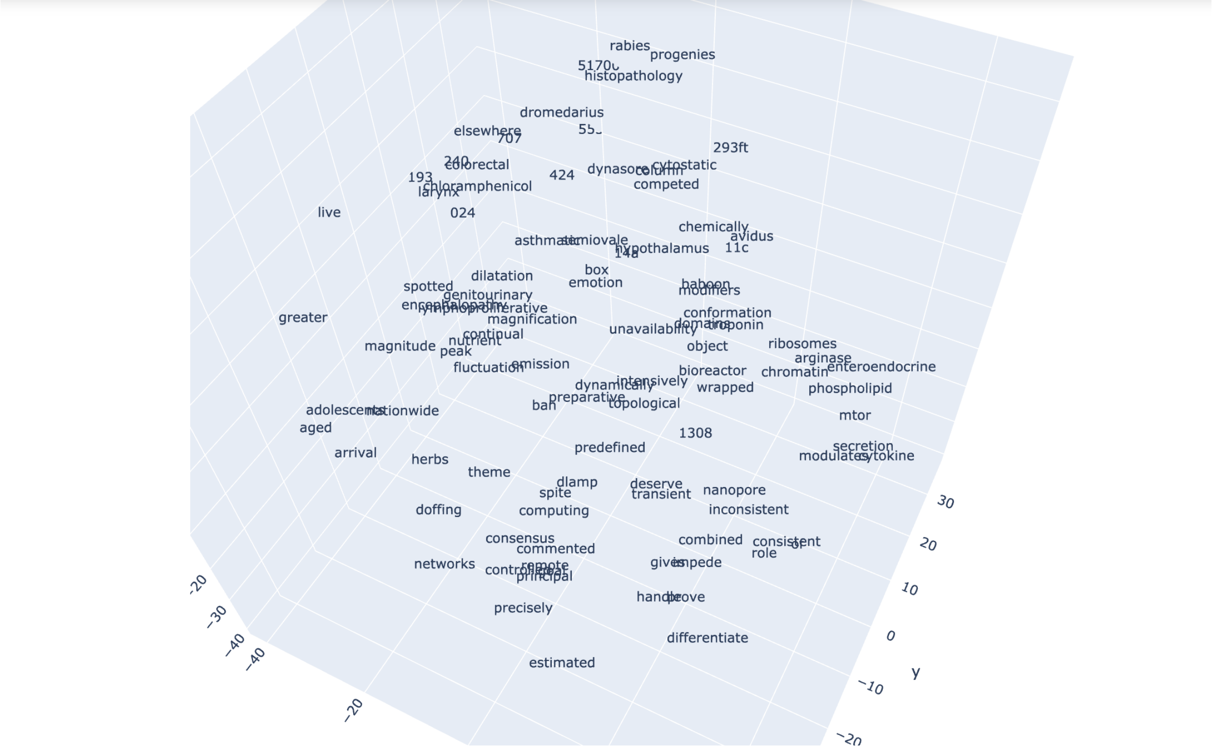
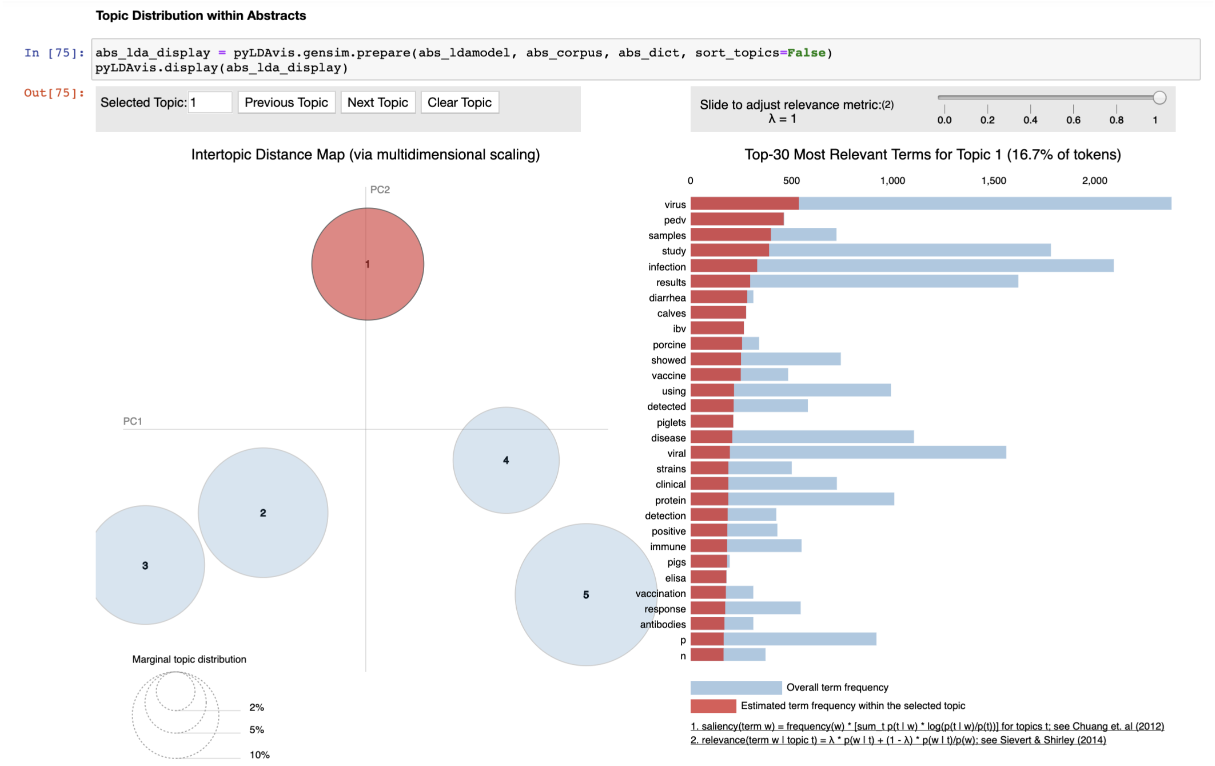


Figure 10: t-SNE of Word Embeddings

Topic Modeling (LDA):

When creating topic models, I wanted to see if papers that had similar proposals in abstracts gave similar results or not. This information can be insightful because if similar proposals gave different results they could be studied to see what were the differences in the methodologies and which paper is providing more promising results. To do this, I created two topic models, one for the abstracts of all papers and the other for the results of all papers. Both the topic models clustered the papers into five topics each.

The pyLDAviz package is used to visualize the topic models. It provides an interactive plot that shows word distribution for each topic. It also plots the topics against 2 principal components, which helps in understanding how similar the topics are to each other. For instance, topic 1 of abstracts contains the words, vaccine, antibody, and animal names which indicates that it probably clusters papers on clinical trials of some vaccines. Topic 5 of results contains similar words to topic 1 of abstracts like treatment, viral, and animal names which indicates that it probably clusters papers on clinical trials too.



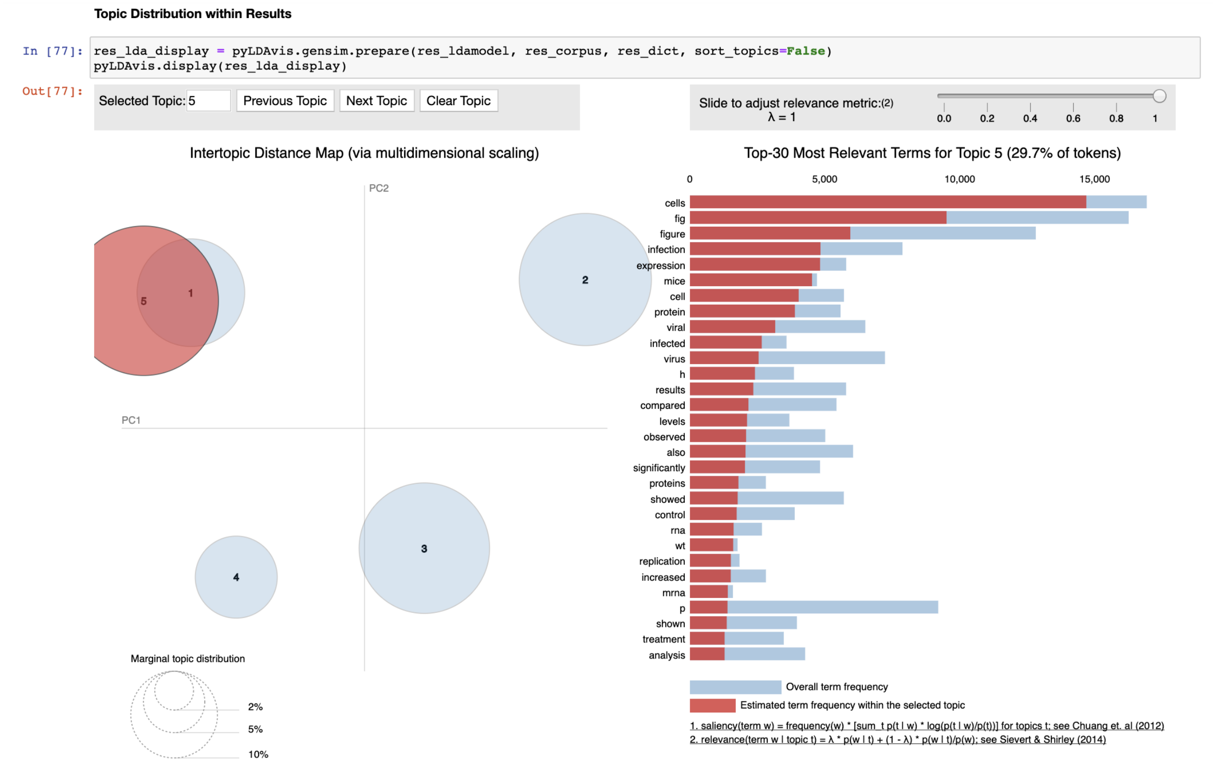
Topic Distribution within Abstracts

```
In [75]: abs_lda_display = pyLDAvis.gensim.prepare(abs_ldamodel, abs_corpus, abs_dict, sort_topics=False)
         pyLDAvis.display(abs_lda_display)
Out[75]:
```

Selected Topic: 1

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 11: Topic Distribution within Abstracts



Topic Distribution within Results

```
In [77]: res_lda_display = pyLDAvis.gensim.prepare(res_ldamodel, res_corpus, res_dict, sort_topics=False)
         pyLDAvis.display(res_lda_display)
Out[77]:
```

Selected Topic: 5

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 12: Topic Distribution within Results

I use a t-SNE plot to see how the topics are clustered across abstracts and results. From the figures we can see that there is some resemblance in the patterns.

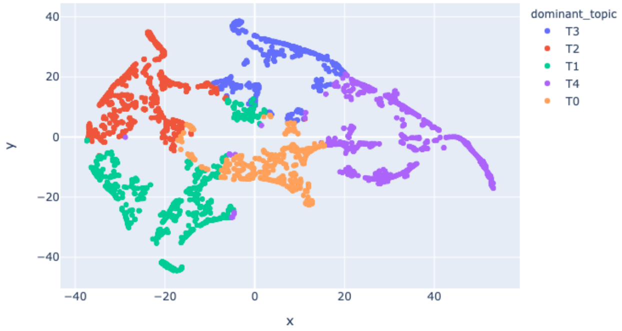


Figure 13: Topic Clusters in Abstracts

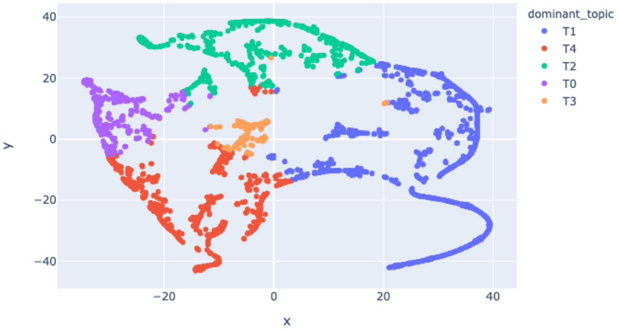


Figure 14: Topic Clusters in Results

But this does not tell us if the same papers are belong to the same cluster between abstracts and results. To check that I calculated several metrics. The most popular cluster evaluation metric is the adjusted rand index. In between the clusters we got from abstracts and results, the rand index was around 0.21 which is quite low.

Since we want to focus on COVID-19 papers, I tried to find which topics have a higher proportion of papers that are tagged as covid_19. This can help researchers focus on papers that are within those topics as they are more likely to be similar to what we are dealing with presently. We can see that topic 4 of abstracts and topic 1 of results have a higher proportion of COVID tagged papers compared to the other topics.

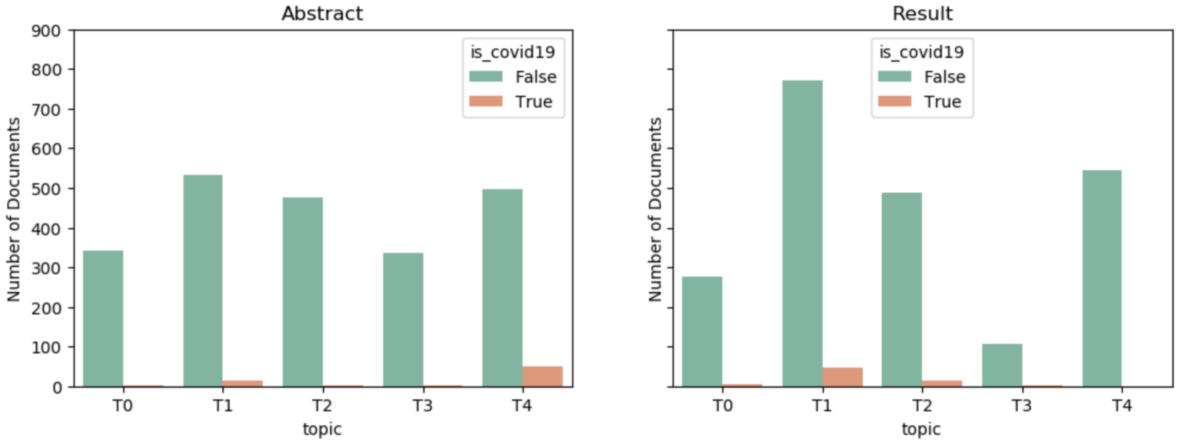


Figure 15: Number of Documents vs Topics grouped by is_covid