

Topic modelling, implementing the model's transformation using vector space algorithms

MICHAEL AJAO-OLARINOYE

DATA SCIENCE AND COMPUTATIONAL INTELLIGENCE

10118047

olarinoyem@coventry.ac.uk

Abstract – this paper was written to show how to use genism to perform topic modelling, using the LDA and LSA and Tf-idf to use for transformation between vector space algorithm. Using Tf-idf model to build the bag of words dictionary(corpus), and build the base model of LDA and also use the tf-idf model to build another LDA model and LSA model. The result shows the performance of the three models which can be found in the code that would be in the appendix.

Keywords – corpus, Latent Dirichlet Allocation (LDA), topic modelling, COVID-19, Latent Semantic Analysis (LSA), Term Frequency * Inverse Document Frequency, (Tf-Idf), PCA

I. INTRODUCTION

In this confusing time that we are in, the year 2020 is a year that would go down in the history books. A pandemic that changes the way humans interact with each other, social norms were broke, the economy shut down and thousands of lives were lost. Coronavirus Disease 2019 (COVID-19) is a severe respiratory illness triggered by a novel coronavirus outbreak. The World Health Organization (WHO) was notified on 31 December 2019 of cases of pneumonia of uncertain microbial aetiology associated with Wuhan City, Hubei Province, China. the virus made its way around the world and infected millions of people.

In case anyone reading this paper is yet to know what COVID-19, The clinical diagnosis is usually one of respiratory failure with disease frequency varying from moderate common cold-like illness to serious bacterial pneumonia progressing to possibly catastrophic acute respiratory distress syndrome (Fauci, Lane, and Redfield 2020). This disease shows us that the rate at which novel coronavirus (Nishiura et al., 2020) evolve is greater than the rate of development of the vaccine.



figure 1 An illustration of people in face mask

The COVID-19 pandemic caused an ongoing surge of discovery, data exchange and accessible science as the medical community try to recognize the epidemic, trace its progress, and examine the SARS-CoV-2 virus that triggers COVID-19. The virus infected a lot of human social norms in different countries. It made people come up and adapt to ways that they were not used to; the rate of transmission is rapid and in an upward trajectory. In figure 1 you can see an illustration of mask-wearing which has become a necessity to stop the spread of the virus.

Topic modelling is a technique in NLP to solve various problems, Topic Models are a form of computational language models used in a set of texts to uncover secret structures. Topic modelling is a classic problem in text mining. Topic modelling is a valuable approach that, in comparison to conventional methods of data reduction in bioinformatics, improves the capacity of researchers to analyses biological knowledge (Liu et al. 2016).

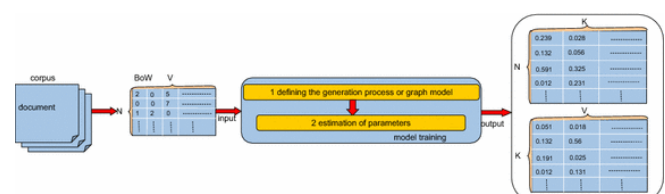


figure 2 The diagram of topic modelling

In this project, I will use topic modelling technique on the COVID-19 research papers dataset. To determine the 5 most popular models in the

This paper contains the following sections which describe what content is available in the sections:

- ## II. DATASET

The dataset used was just a part of the full dataset, the dataset was called metadata.csv. CORD-19 is a dataset containing over 200,000 scientific publications, covering over 94,000 full-text papers on COVID-19, SARS-CoV-2, and associated coronaviruses. In this paper, we will only use the ones that have an abstract and the papers that were

figure 3 The graphical representation of the LDA model

B. Latent Semantic Analysis (LSA):

LSA utilizes a word model container, which results in a term-domain matrix comprising words in a statement. LSA studies latent topics by executing a matrix decomposition on a document-term matrix using a single decomposition meaning. LSA is usually used as a spatial or noise-reducing tool. SVD is a matrix factorization process that describes a matrix in the form of two matrixes. LSA which is also known as LSI (Latent Semantic Index).

C. Term Frequency * Inverse Document Frequency, (Tf-Idf):

Term Frequency * Inverse Document Frequency is a numerical statistic meant to represent the value of a term in the set or corpus of a document. This is commonly used as a weighting element in information retrieval queries (Wikipedia Contributors 2020). The Tf-idf is the combination of two factors, the word frequency and the inverse frequency of documents. It is expecting a bag-of - word (integer values) testing corpus during initialization. Through the transition, a vector will be used and another vector with the same dimensionality will be recovered so that features that were uncommon in the testing corpus would raise their meaning. It then transforms integer-valued vectors into real-valued vectors, keeping the number of dimensions unchanged. Tf-idf is a big part of the published research as it is both a corpus discovery tool and a pre-processing stage for several other text-mining steps and models (Lavin 2019).

IV. EXPERIMENTATION AND RESULT DISCUSSION

For the analysis that would be done in this paper the python package that will be used for building the LDA and some preprocessing is called genism. We will be using google colab as the development environment as there is access to virtual GPU processing power that will aid in the analysis.

The experimentation will take place in the following form(pipeline):

- Install and load all the libraries and dependencies needed
- Load the data
- Perform analysis on the data by reducing it to the amount of metadata that has abstract

- Preprocess the data by performing text processing, tokenization, remove stop words, lemmatization, and stemming
- Transform the data into a vectorized form
- Build the two models.

Installing and loading all the libraries and dependencies which include pandas, NumPy, genism, nltk, sklearn PCA, seaborn, wordcloud and matplotlib. Which will assist us in preprocessing the data, building the model, analyzing the performance of the models and the topic visualization.

Loading the data and filtering through all the papers in the dataset and selecting only the papers that were published after 1st January 2020. Which resulted in the figure given below from 215527 papers that were in the data in total.

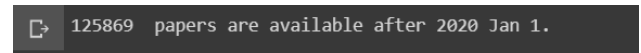


figure 4 Result of the amount of paper after filtering

From this we must filter to reduce it to the papers that have abstract included in the papers, the figure below shows the result generated after running the code to find the paper that has abstract written in them. After this result, the data was passed into a pandas dataframe for preprocessing.

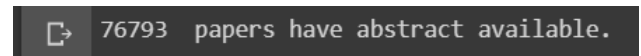
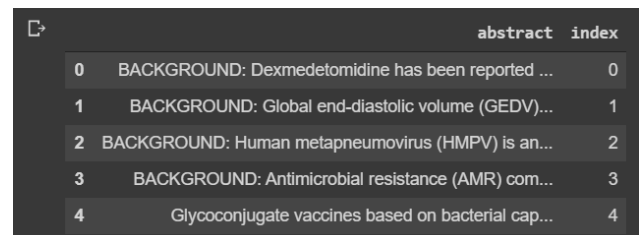


figure 5 Result after filtering

The dataframe was preprocessed by performing text processing, tokenization, remove stop words, lemmatization, and stemming, python functions were written to perform the following that was listed above.



	abstract	index
0	BACKGROUND: Dexmedetomidine has been reported ...	0
1	BACKGROUND: Global end-diastolic volume (GEDV)...	1
2	BACKGROUND: Human metapneumovirus (HMPV) is an...	2
3	BACKGROUND: Antimicrobial resistance (AMR) com...	3
4	Glycoconjugate vaccines based on bacterial cap...	4

figure 6 An example of the dataframe to be preprocessed

Tokenization is the act of breaking down a text into words, punctuation marks, numeric digits. Stemming refers to reducing a word to its root form. Lemmatization reduces the word to its root, as it appears in the dictionary. The preprocessing step taken will be found in the code link which will be included in the bottom of the paper

```
Original document:
['OBJECTIVES:', 'To', 'describe', 'experience', 'with', 'airway', 'pressure', 'release', 'ventilat

Tokenized and lemmatized document:
['objective', 'experience', 'airway', 'pressure', 'release', 'ventilation', 'aprv', 'child
```

figure 7 The result before and after tokenization and lemmatization

The figure above shows an example of the of a document that was preprocessed using the steps discussed in this paper.

Let us focus on converting text data into a format that will act as an input for the LDA model training. We continue by converting the documents to a simple vector representation. To create a bag of words for the dataset, a function was created to build a dictionary using the preprocessed documents, also removing the very extreme and very common words by filtering through the dictionary of words that were created. A bag of word model was created for each document and was called bow_corpus.

```
[ ] #create tf-idf from bow_corpus
tfidf = models.TfidfModel(bow_corpus)
corpus_tfidf = tfidf[bow_corpus]
```

figure 8 Creating the Tf-idf corpus using transformation

Build Tf-idf model object using models.TfidfModel on 'bow_corpus' and transfer it to 'tfidf,' then apply a transformation to the entire corpus and label it 'corpus tfidf' which is the second model and it will be called corpus_tfidf. So, there are two models using LDA + bow_corpus and LDA + corpus_tfidf which was trained using the gensim.models.LdaMulticore. The first model which is the base model was trained with the LDA multicore algorithm + the bag of words with 5 number of topics to be found to make 50 passes, the passes make it that the algorithm would find a more accurate topic. In the figure below we can see the algorithm built with hyperparameters and the time it took for the algorithm to run.

```
[ ] now = datetime.datetime.now()
print ("start model building at ",now.strftime("%Y-%m-%d %H:%M:%S"))
lda_model = gensim.models.LdaMulticore(bow_corpus,
                                       num_topics=5,
                                       id2word = dictionary,
                                       passes = 50,
                                       workers=10)

now = datetime.datetime.now()
print ('Model training finished at ',now.strftime("%Y-%m-%d %H:%M:%S"))

start model building at 2020-08-16 19:07:10
Model training finished at 2020-08-16 19:40:33
```

figure 9 The base LDA model

For each topic, we will investigate the words that occur in the topic and their relative weight. From the result gotten from the topic, we can see that the figure below which is a chart of shows the words that are in 4 topics.

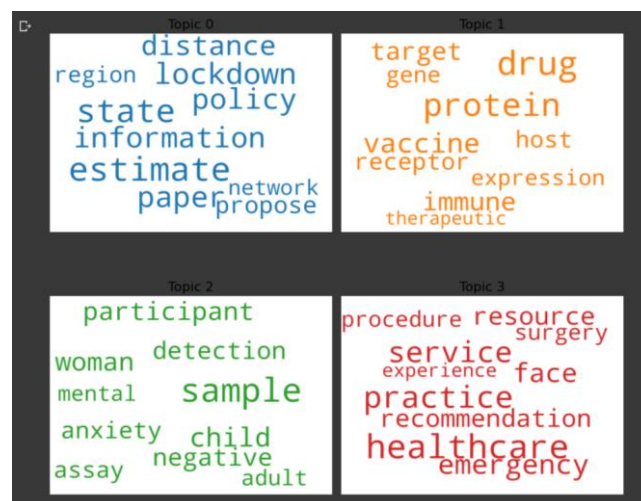


figure 10 The result of the word of 4 topics for the base model

The second algorithm was also built with the same process and parameters and the time it took to run was determined. The corpus_tfidf which was built with the Tf-idf model.

```
[ ] now = datetime.datetime.now()
print ("start model building at ",now.strftime("%Y-%m-%d %H:%M:%S"))

lda_model_tfidf = gensim.models.LdaMulticore(corpus_tfidf,
                                              num_topics=5,
                                              id2word = dictionary,
                                              passes = 50,
                                              workers=10)

now = datetime.datetime.now()
print ('Model training finished at ',now.strftime("%Y-%m-%d %H:%M:%S"))

start model building at 2020-08-16 19:40:33
Model training finished at 2020-08-16 20:26:44
```

figure 11 The model of the tf-idf model using the tf-idf corpus

The figure above shows the code written to train the second model running LDA with the tf-idf corpus model built

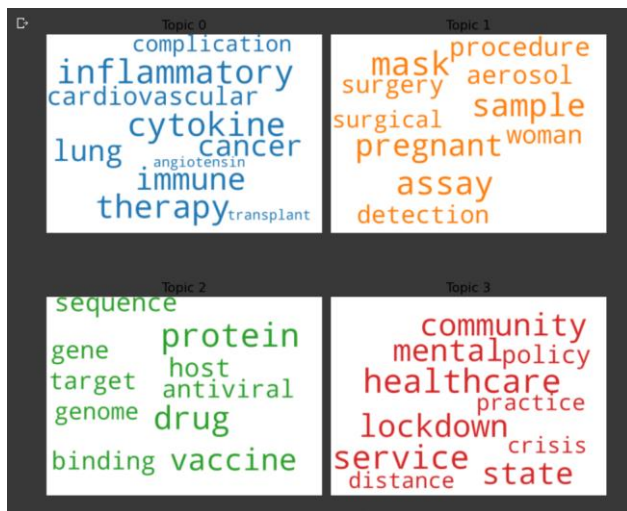


figure 12 The result of the words of 4 topics for the Tf-idf model

The figure above shows the result gotten from the second algorithm that was built with the corpus_tfidf model from the looks at the topics that were produced we can see that the result gotten from both models are different.

```
now = datetime.datetime.now()
print ("start model building at ",now.strftime("%Y-%m-%d %H:%M:%S"))

lsi_model_tfidf = gensim.models.LsiModel(corpus_tfidf,
                                         num_topics=5,
                                         id2word = dictionary,
                                         )

now = datetime.datetime.now()
print ('Model training finished at ',now.strftime("%Y-%m-%d %H:%M:%S"))

start model building at 2020-08-23 18:51:04
Model training finished at 2020-08-23 18:51:33
```

figure 13 The LSA model with Tf-idf corpus

The LSA model, the figure above shows the python code written to build the model that uses the tf-idf corpus dictionary. The image below shows the words from the topics.



figure 14 The result of the words of 4 topics for the tf-idf model

LDA + bow_corpus: topic probability:						
LDA + TF-IDF_corpus: topic probability:						
	topic1	topic2	topic3	topic4	topic5	final_topic
0	0.000000	0.0	0.000000	0.000000	0.992304	topic5
1	0.429467	0.0	0.000000	0.055057	0.511607	topic5
2	0.000000	0.0	0.837611	0.000000	0.154978	topic3
3	0.000000	0.0	0.363263	0.525348	0.108424	topic4
4	0.000000	0.0	0.984217	0.000000	0.000000	topic3

figure 15 Topic probability of the 2 models

Based on max-probability we pick the final topic for each abstract gotten from the two models. The image above shows the dataframe of the two models. To understand the performance.

After running the algorithms, the next step is to visualize the result gotten, I will be using the model built with the Tf-idf transformation. We used PCA-2D, PCA-3D, and T-SNE to visualize the distribution of topics in all abstracts. the image below shows the PCA-2D of the model. The visualization process was taken using PCA, the explained variance for the component ratio is 3. The 3 PCA is then visualized in the 2D form. Each colour represents the five topics and the variance explained. A dataframe was created from the projected vectors for the PCA.

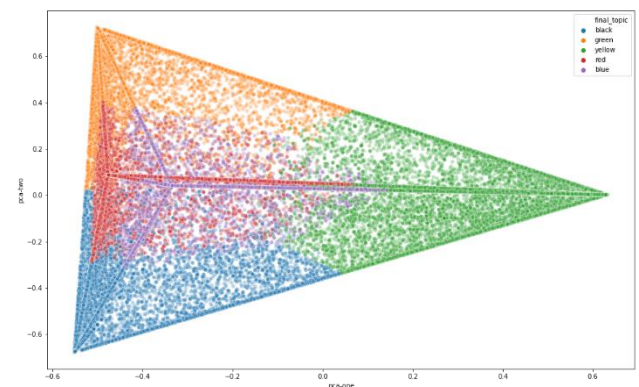


figure 16 2D-PCA representation of the 5 topics chosen

The image below is a 3D representation of the 5 topics in the 3 PCA components, the distribution 3D of the topics that was built with the Tf-idf model of the corpus, three models were built, the base LDA model, the Tf-idf LDA model and the Tf-idf LSA model.

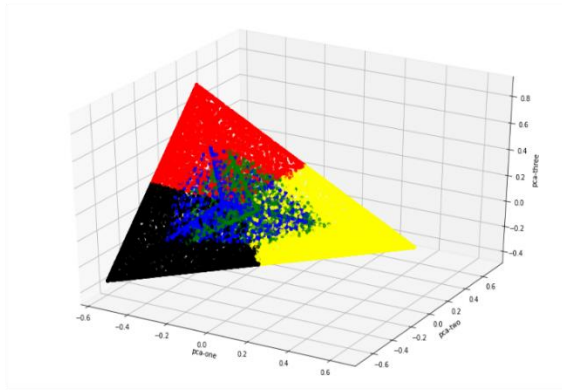


figure 17 3D-PCA representation of 5 topics chosen

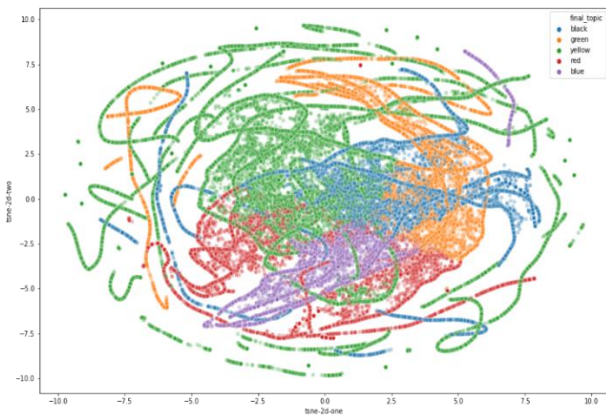


figure 18 Report clusters in 2D space using t-SNE (t-distributed stochastic neighbour embedding)

V. CONCLUSION

The conclusion of this paper is to show that transformation models produce different results for topic modelling. Two models were used in transforming vectors because vector space models can be transformed between themselves. LDA and LSA which can be said to be LSI were used in this paper, while Tf-idf was used to transform the corpus dictionary and the result gotten was visualized using PCA both in 2D and 3D. The use of a python library to solve the problem of topic modelling and the use of genism which is a python package show how we can utilize with topic modelling. The results show the five topics that were built with the model.

VI. APPENDIX

<https://colab.research.google.com/drive/1-13cnYPwTjjA6jl14DezKrKvnnDzKYwl?usp=sharing>

I acknowledge inioluwa Temitope Akande as we both work on this project together

VII. REFERENCES

- Lavin, M.J. (2019) Analyzing Documents with TF-IDF [online] Available from <<https://programminghistorian.org/en/lessons/analyzing-documents-with-tfidf>> [16 August 2020]
- Fauci, A.S., Lane, H.C., and Redfield, R.R. (2020) “Covid-19 — Navigating the Uncharted”. New England Journal of Medicine [online] 382 (13), 1268–1269. available from <<https://www.nejm.org/doi/full/10.1056/nejme2002387>> [15 August 2020]
- Liu, L., Tang, L., Dong, W., Yao, S., and Zhou, W. (2016) “An Overview of Topic Modeling and Its Current Applications in Bioinformatics”. SpringerPlus [online] 5 (1). available from <<https://springerplus.springeropen.com/articles/10.1186/s40064-016-3252-8#citeas>> [16 August 2020]
- Nishiura, H., Jung, S., Linton, N.M., Kinoshita, R., Yang, Y., Hayashi, K., Kobayashi, T., Yuan, B., and Akhmetzhanov, A.R. (2020) ‘The Extent of Transmission of Novel Coronavirus in Wuhan, China, 2020’. *Journal of Clinical Medicine* [online] 9 (2), 330. available from <http://dx.doi.org/10.3390/jcm9020330>
- Hui, J. (2019) Machine Learning — Latent Dirichlet Allocation LDA - Jonathan Hui - Medium [online] Available from <https://medium.com/@jonathan_hui/machine-learning-latent-dirichlet-allocation-lda-1d9d148f13a4> [16 August 2020]
- Blei, D., Edu, B., Ng, A., Jordan, M., and Edu, J. (2003) “Latent Dirichlet Allocation”. *Journal of Machine Learning Research* [online] 3, 993–1022. available from <<https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>> [16 August 2020]
- Wikipedia Contributors (2020) Tf-Idf [online] Available from <<https://en.wikipedia.org/wiki/Tf%E2%80%93idf>> [16 August 2020]

