Interactive meta-cluster detection in Twitter data

Juan Antonio Ramirez-Orta
Department of Computer Science
Dalhousie University

Muthukumar Rajendran
Department of Computer Science
Dalhousie University

Suraj Kandikonda
Department of Computer Science
Dalhousie University

Yaswanth Chiruvella
Department of Computer Science
Dalhousie University

Fall 2020

Abstract

Social media has been the main online mean of interaction among individuals during the COVID-19 pandemic. People are increasingly using social media, especially online communities, to discuss health concerns and seek for support. Understanding topics, sentiment and structure of these communities could potentially reveal important aspects of health-related conditions or psychological or sociological trends. In this project, we investigate and identify latent meta-groups of topics and trends with mental health-related conditions including depression and anxiety, using a large dataset extracted from Twitter. To understand the data, we performed Unsupervised Machine Learning techniques on the sentence embeddings obtained using Deep Language Models, and then we analysed the topic-based features present in the posts made by members in those clusters. The work focuses on providing tools to cluster and interpret the sentence embeddings provided by the Language Models using classical Machine Learning and Natural Language methods. We also propose the use of a type of Graph and some basic Community detection algorithms inside an interactive tool to involve the user in the process of topic discovery.

1 Introduction

Social media sites such as Facebook, Twitter, Reddit and Tumblr have become increasingly recognized as promising platforms to understand the psyche of populations. People have been moving away from their traditional communication, now meeting up online and using social media tools to make a different communication model. Online communities have been built up as forums for individuals to share information and advice in a variety of their daily life, especially their health beings. With such popularity, social media offers a low-cost sensing channel to analyse health behaviours of individuals and communities through their postings. Nonetheless, the vast amount of content generated by the users imposes a challenge on the analysis, which can be addressed using techniques from Natural Language Processing (NLP).

The standard NLP technique to summarize large collections of documents is called Topic Modelling, which has as its most important representative the Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. The purpose of Topic Modelling is to find groups of words (called topics) that can explain the observed distributions of the words across the document collection.

On the other hand, with the recent developments on Representation Learning and Pre-trained Language Models such as BERT [Devlin et al., 2018], there has been significant progress into unified, off-the-shelf, general-purpose representations for text (called embeddings) that allow for quantitative analysis of text data.

In this work, we combine LDA, sentence embeddings and Graph Visualization techniques in an interactive app to enable the user analyze large amounts of text. Our contributions of this work are: (1) a novel general framework to visualize clusters of sentence embeddings and (2) a visualization tool that helps to view the topic clusters interactively.

2 Dataset

For this project, we use a data set from Kaggle [Kaggle, 2020b] which has about 6.5 million records starting from March 2019, which can be found here [Kaggle, 2020a]. Some basic descriptive statistics are shown in table 1. This data set has 7 columns: ID, Username, location, URL, hashtag, time and month. These columns help us in manipulating the data in a timely basis. We could see what tweet is posted and in which month it was posted and the hashtags that are associated with that tweet. To be able to process the dataset, we took a random sample of 10,000 tweets. We found this to be the biggest number of tweets that is both interesting and tractable for most laptop hardware.

3 Pre-trained language models

With the recent advent of Deep Learning techniques across several areas of Computer Science, there has been significant effort in finding mathematical

Number of metadata attributes	7
Number of tweets	6,556,683
Total size in memory	2.2 GB
Average tweet size in memory	1KB
Unique Users	85574
Time Frame	March 8,2020 - April 24, 2020

Table 1: Profile of the original Twitter dataset. Metadata includes ID, Username, location, URL, Hashtag, time and month. For demonstration purposes, we only processed 10,000 tweets.

representations for text that encode the semantics of language. The basic idea is to create Neural Networks that learn fixed-length vectors for words (which are called word embeddings) that represent their meaning and then use these word embeddings to identify properties of the words or to combine them to produce sentence embeddings or document embeddings.

The first approaches for producing off-the-shelf, general-purpose word embeddings were Word2Vec [Mikolov et al., 2013], GloVe [Pennington et al., 2014] and ELMo [Peters et al., 2018], but in 2018, BERT [Devlin et al., 2018] set a new standard in the Natural Language Processing community because its embeddings allowed the creation of several systems that achieved a new state of the art performance in many tasks of the GLUE benchmark [Wang et al., 2018]. Although the performance of BERT is great, it has two major disadvantages: the first one is the computational requirements to use it, and the second one is that it doesn't directly produce appropriate sentence embeddings. To address these issues, we used SDistilBERT [Reimers and Gurevych, 2019], an adapted version of DistilBERT [Sanh et al., 2019] for the task of Semantic Textual Similarity, which is the task of produce meaningful sentence embeddings. The sentence embeddings produced by SDistilBERT are suitable for analysis with standard Machine Learning algorithms, as they are meant to exploit the geometrical properties of Euclidean space to encode meaning.

4 Agglomerative clustering and dendrograms

Agglomerative clustering is an unsupervised machine learning algorithm that combines similar objects into groups called clusters. The output of the algorithm is a set of clusters, where the objects inside each cluster are similar with each other and different from the objects in the other clusters. Hierarchical clustering starts by treating each observation as an individual cluster, and then it iteratively merges the two most similar clusters until all the data points belong to the same group. At every iteration of the algorithm, to decide which two clusters are the most similar (and therefore merge them), an aggregate measure of the distance between clusters is needed, which leads to the concept of the linkage hyper-parameter. There are several options for the linkage: single (where

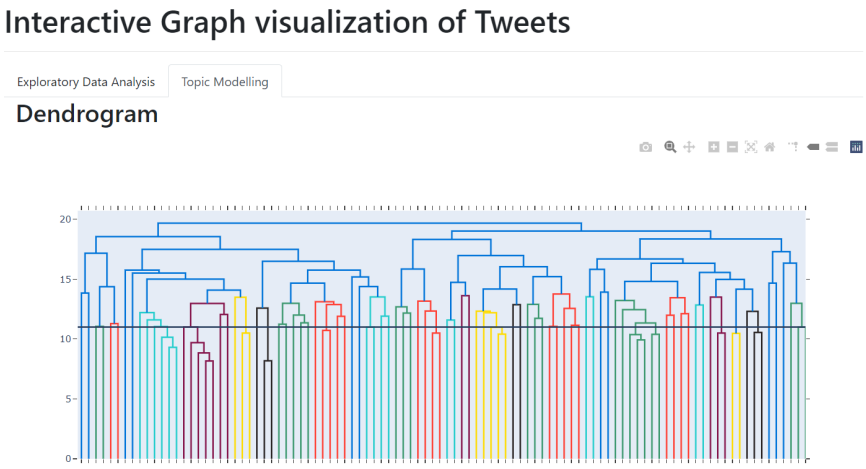

Figure 1: A dendrogram representing an Agglomerative Clustering. The vertical lines represent clusters, while the horizontal lines represent merging of the clusters. In the beginning of the algorithm, every node is a cluster of one element (displayed at the bottom), and as the iterations pass by, the clusters are merged together until only one cluster is formed (displayed at the top).

the distance between clusters is defined as the minimum distance between their elements), complete (where the distance between clusters is defined as the maximum distance between their elements) and mean (where the distance between clusters is defined as the mean of the distances between their elements).

To visualize an Agglomerative Clustering, the standard method is the Dendrogram, like the one shown in figure 1. It is similar to a tree, where the clusters are represented as the nodes in the tree and they are being merged as the iterations of the algorithm pass by. It is usual to have the leaf nodes or the child nodes present in the bottom most while the root node or the parent nodes are in the top most.

Two important disadvantages of the dendrogram visualization is that is it not easy to identify the clusters being formed and that visualization does not provide any kind of useful information about what is present inside the clusters. Users can find the number of clusters present in the data set and can notice how the data is being traversed from the root to the leaf node, but this does not help the user to derive meaningful insights right away. For instance, if the user would like to complement the Agglomerative Clustering with other techniques, it is difficult to understand how the additional techniques interact with the Agglomerative Clustering.

5 Topic Modelling

A classical approach to summarize large amounts of text is Topic Modelling, a branch of unsupervised Natural Language Processing (NLP), in which we try to find groups of words (called the Topics) that appear jointly and use this groups to describe large collections of documents. These topics try to better explain the hidden information present in the collection of documents. There are many methods for performing the topic modelling over the data, but we chose the general probabilistic approach called Latent Dirichlet Allocation [Blei et al., 2003], which has been the state of the art method for around a decade.

LDA is a generative probabilistic model that can discover the latent semantics embedded in document corpus and the semantic information can be much more useful to identify the document groups than raw term features. LDA's approach to topic modelling is to consider each document as a collection of topics in a certain proportion and each topic as a collection of keywords, again, in a certain proportion. When we provide the algorithm with the number of topics, all it does it to rearrange the topics distribution within the documents and keywords distribution within the topics to obtain a good composition of topic-keywords distribution. Here, a topic is nothing but a collection of dominant keywords that are typical representatives. Just by looking at the keywords, we can identify what the topic is all about. The following are key factors to obtaining good segregation topics: the quality of text processing, the variety of topics the text talks about, the choice of topic modelling algorithm, the number of topics fed to the algorithm, the algorithms tuning parameters.

The basic hypothesis of LDA is that Topic Modelling can be cast as a distribution of distributions. More specifically, a collection of documents is characterized by its distribution of topics in the documents and the distribution of words en every topic. Because of this, LDA factors the Word-Document matrix into two matrices, which are the probability distribution of topics in documents and the probability distribution of words in topics. Normally, in classic document clustering approaches, documents or posts are represented with a bag-of-words (BOW) model which is purely based on raw terms and it is insufficient to capture all the semantics.

Despite its success, LDA has some serious disadvantages: the number of topics has to be selected manually, it does not take into account word embeddings from pre-trained language models, it is hard to justify the Dirichlet prior even when it is mathematically convenient and the topics found don't reflect well-known properties of text (e.g. Zipf's Law [Zipf, 1949]). A survey of LDA and its alternatives can be found in [Blei, 2012].

Recently, much attention has been paid to visualizing the output of topic models such as LDA. Such visualizations are challenging to create because of the high dimensionality of the model. LDA is typically applied to many thousands of documents, which are modeled as mixtures of dozens (or hundreds) of topics, which themselves are modeled as distributions over thousands of terms. The most promising basic technique for creating LDA visualizations that are both

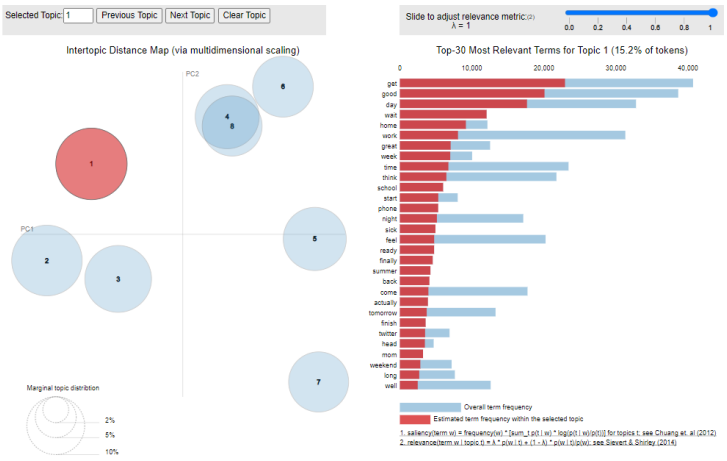

Figure 2: The pyLDAvis system. It allows the user to select the number of topics, compare them and tune the fitting algorithm, but the user is not able to directly interact with the documents present in the topics. Our proposed system overcomes this limitation.

compact and thorough is interactivity.

One of such interactive tools is pyLDAvis [Sievert and Shirley, 2014], which is a visualization tool that helps to visualize the topics that are generated using LDA. This tool helps in visualizing the topic clusters and the most frequently used terms in a particular cluster, but it doesn't provide details about the data inside and it limits the way in which the user interacts with the system. The user can only choose the topic clusters and they able to see only the frequent tokens in that cluster. A sample screenshot of the pyLDAvis system is shown in figure 2.

6 Text preprocessing

To make the data suitable for Topic Modelling, the first step we did was to perform the frequency counts of unigrams, bigrams and trigrams. We performed this counts with the standard library Gensim [Řehůřek and Sojka, 2010], which has as the two most important hyper-parameters the min-count and threshold, which control the minimum occurrences needed for a pair of words to be considered as an important n-gram: the higher the values of these hyper-parameters, the harder it is for words to be combined into n-grams.

The second step is to create the two main inputs of the LDA topic model, which are the dictionary (id2word) and the corpus. The dictionary is created using the Gensim's [Řehůřek and Sojka, 2010] pre-defined method, which has as inputs the lemmatized words present in the corpus. Gensim creates a unique

id for each word in the document, so that the produced corpus is a mapping of word-id and word-frequency. The corpus is simply the list of documents (text strings) present in the collection.

The final step is to fit the model. In addition to the corpus and the dictionary, the desired number of topics is needed as well. Apart from that there are other hyper-parameters that we left untouched: alpha and eta are hyper-parameters that affect sparsity of the topics, and according to the Gensim [Řehůřek and Sojka, 2010] docs, both default to 1, the number of topics prior; the chunk size is the number of documents to be used in each training chunk; the update value determines how often the model parameters should be updated and the passes is the total number of training passes for the model to fit.

7 Visualizing dendrograms as Graphs

To understand the relationships between the clusters produced by the Agglomerative Clustering, we propose to visualize them as a Graph, which is computed in two steps. The first step is to produce an aggregate representation for each cluster as the mean of the representations inside the cluster. The second step is to compute all the pairwise distances of the aggregate representations of the clusters to form a matrix M, given by the formula $M_{i,j} = d(C_i, C_j)$, where C_i and C_j are the representations of the clusters i and j, respectively, and $d(,)$ is the same distance function used to perform the Agglomerative Clustering. It is easy to see that the matrix M is non-negative, symmetrical and has only zeros in its diagonal, so that M defines a complete Graph over the considered clusters.

To make it possible to visualize this Graph, we make the observation that by only plotting the lowest values in the matrix M, we can retain the most meaningful connections, and the best way to decide which is the most appropriate level of detail is by involving the user in the graph pruning process.

8 Proposed system

As far as we know, currently there is no system that combines both Topic Models and Deep Language Models in an interactive visualization to describe textual data. To get the best from both approaches, we developed a tool that allows the user to perform Topic Modelling and Agglomerative Clustering on sentence embeddings in an interactive manner. In our interactive system, there are two tabs: the first one is to perform a basic exploratory analysis of the data and the second tab is to perform a more detailed analysis involving Topic Modelling and Agglomerative Clustering.

The first tab (shown in figure 3) has the following features: a line plot to describe the amount of tweets across time, a word cloud showing the most frequent hashtags present in the dataset and a word cloud showing the most frequent unigrams, bigrams and trigrams present in the dataset. This tab helps the user to gain an initial insight with very simple and interpretable models,

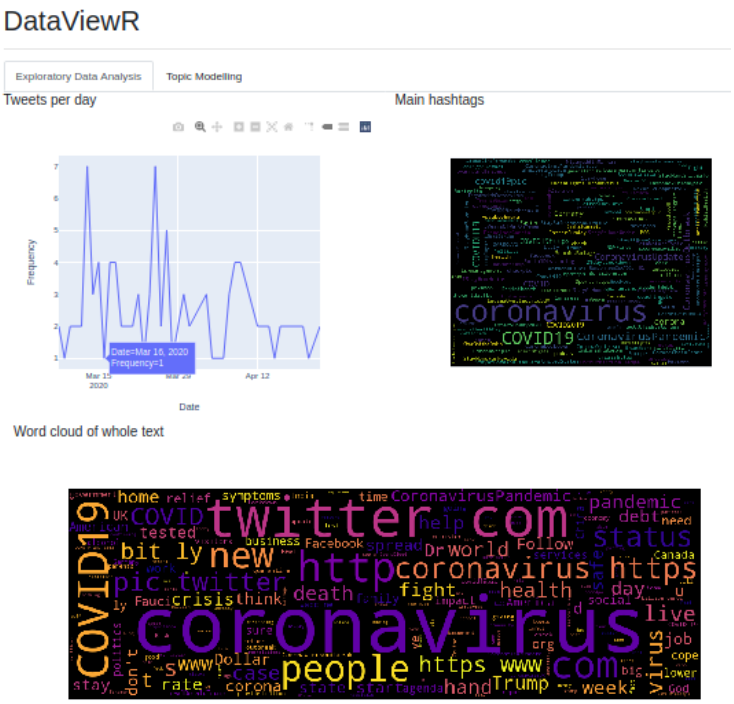

Figure 3: The first tab of the system, where a basic analysis of the dataset is performed. It has the following features: a line plot (top left) that displays the number of tweets posted in each day from March 2020 to April 2020, a word cloud (top right) that displays the most prevalent hashtags in the whole dataset, a world cloud (middle) that displays the most important words in the whole dataset and a bar chart (botton) that displays the topics found using LDA in the whole dataset, along with their relative strength.

with the objective to narrow down the path on what they are looking into and get an insight about what is being discussed before performing any machine learning method.

In the second tab (shown in figure 4), a more advanced analysis can be performed. The user can interact with the system and view the tweet clusters formed by the Agglomerative Clustering of the sentence embeddings in the form of a Graph, where the nodes are the clusters and there is an edge between the nodes if the distance between the centroids is less than a certain threshold, which is selected by the user. This tab has the following features: there is a dendrogram to describe the evolution of the Agglomerative Clustering, a histogram to describe the distances between the sentence embeddings of the individual tweets, a histogram to describe the distances between clusters, a Graph visualization of the clusters with their relationships and two cards to describe

the topics present in each cluster and to browse directly the tweets that compose them. All the visualizations are interactive, so that the user can adjust the values that control the formation of edges and clusters using two sliders: one to select the iteration of the Agglomerative Clustering and one to control the amount of edges displayed in the Graph visualization. Once after the clusters have been formed, the user can click on any cluster to perform LDA on that particular cluster and the topics generated through LDA are visualized through the first card, while the user is able to browse the tweets in the second card.

9 Functionality

In this section, we discuss the minimum, expected and the bonus functionalities that we achieved in this project, along with some details about the software used and the feedback we received.

9.1 Feedback and improvements

After our meeting with Dr. Paulovich and the demo we gave as part of the class presentations, we received several suggestions to improve our original proposal. This suggestions were: adding word clouds in the first tab, display the dendrogram in the second tab, show the tweets inside the clusters, visualize clusters of tweets instead of individual tweets, remove the axes in the word clouds, put meaningful labels for the line plot in the first tab, improve the stop word detection in the word clouds, place our plots as a single dashboard to prevent the user from losing context, show the topics of the clusters on top of the graph and show the topic strengths as a bar chart in the first tab. We were able to successfully implement all this changes, and they dramatically improved the readability and capabilities of our system.

9.2 Minimum and Expected functionality

In our original proposal, our main aim was to develop an interactive interface enabling the user to do three major operations, which were: enabling the user to use their own Twitter data, enabling the user to choose the edge threshold value using a sliding bar and enabling the user to select multiple hashtags that are retrieved through the model. As agreed with the professor, we decided to drop the first requirement and instead focused on making the engine faster and deriving insights from the data we found. We were able to completely fulfill this new minimum functionality as shown in figure 5. Additionally, we provide two sliders for the users to adjust the edges between each node and the cluster size to dynamically compute the Graph, fulfilling the Expected Functionality of our proposal. This helps the user to view the clusters that are closely related and how the clusters are distinct from one another. The slider that adjusts the number of clusters helps the user to know the dominant topics that are present in the data. The user can understand the meaning of the edge and the

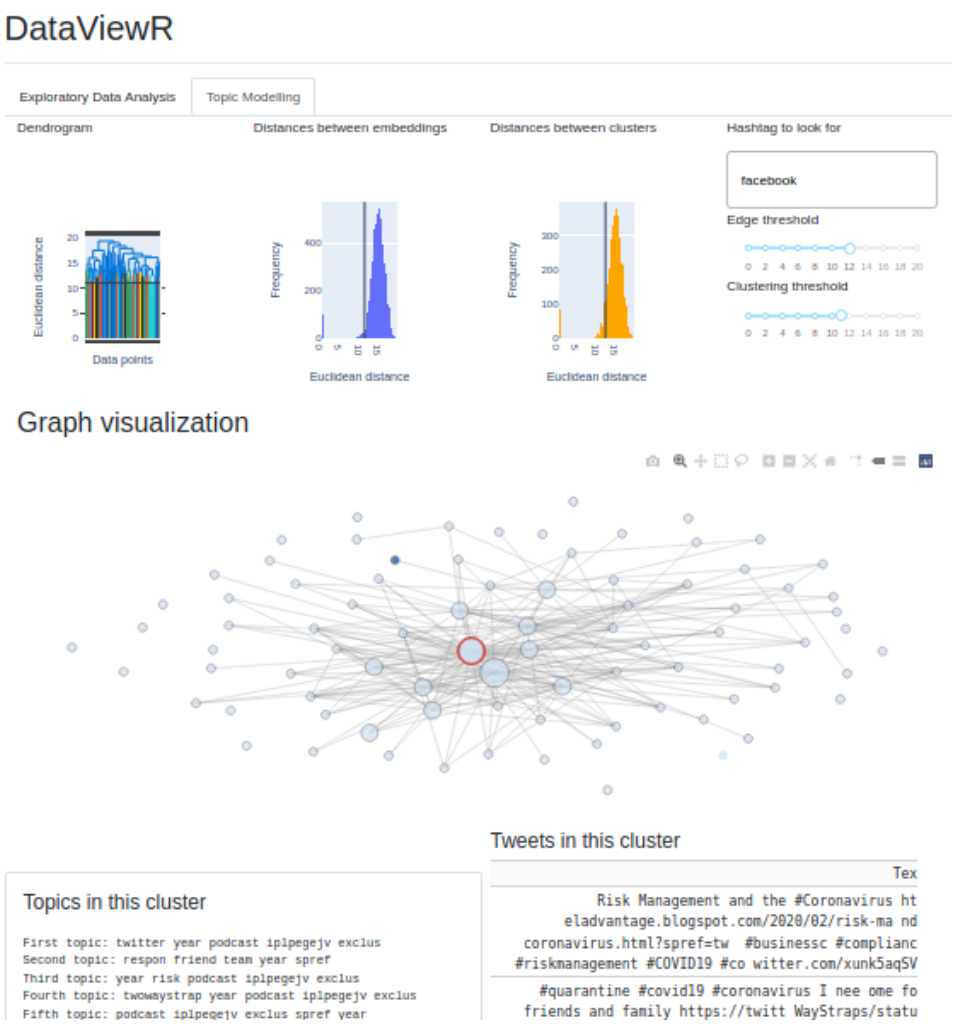

Figure 4: The second tab of the system, where a more advanced analysis can be performed. It has the following features: a dendrogram (top most) to visualize the process of the clustering, a histogram (blue, top left) to describe the distances between individual tweets, a histogram (orange, top right) to describe the distances between the clusters at the current iteration of the algorithm, a text box (middle left) to look for hashtags in the cluster, two sliders (middle right) to control the iteration of the clustering and the edges to display in the graph, a graph visualization (middle bottom) where the user can interact with the clusters being formed and obtain basic information about them and two cards (bottom most) that display the topics present in the current along with the individual tweets upon clicking.

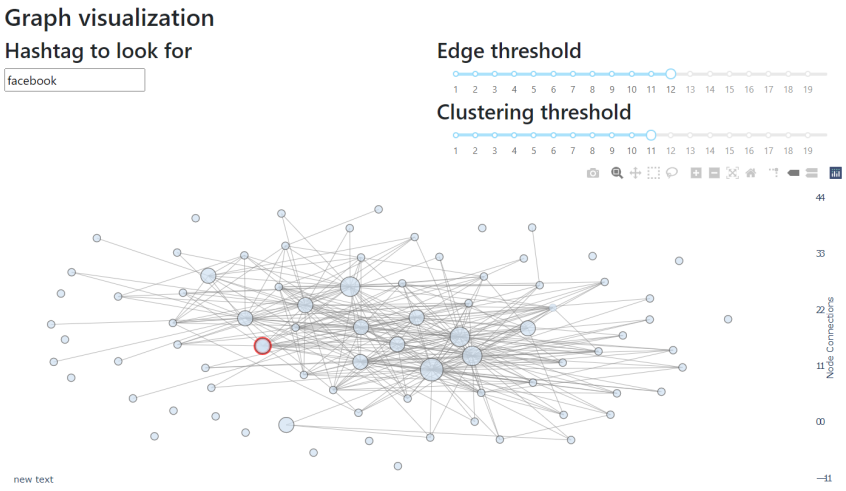

Figure 5: Minimum functionality of the system. It has two sliders to control the number of clusters and the number of edges between the nodes and a search box to look for specific hashtags. As the user changes these parameter the Graph changes.

clustering thresholds using the histograms. This helps the user to choose the best parameters for maximum intepretability.

We developed the entire application in Python, using Dash [Dash, 2020], sklearn [Pedregosa et al., 2011], NetworkX [Hagberg et al., 2008] and sentence-transformers [Reimers and Gurevych, 2019]. After achieving the proposed expected functionality, we were able to derive insights by using our system on the data set. It is obvious that the coronavirus has been a major topic in Twitter and additionally we found that there are topic clusters that are related to mental wellness like depression and anxiety as well. This proves that our system shows clusters that are distinct from one another and it also visualizes how closely these clusters are related to one another using the edges between the nodes in the graph.

9.3 Bonus functionality

Since we were able to implement the expected functionality, we decided to explore our proposed bonus features. In order to gain more insight about the clusters found, we decided to display fine-grained details of the clusters upon hovering and clicking by the user. The first bonus feature is that when the user hovers on a node of the graph, the hover text displays the size of the cluster along with the most frequent hashtags of the cluster, as depicted in figure 6. The second bonus feature we implemented is that when the user clicks on a node of the graph, a fast version of LDA is performed to inform the user of the main top-

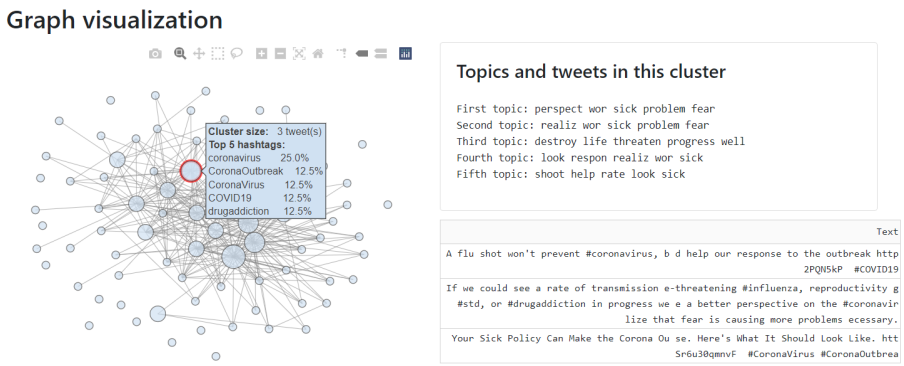

Figure 6: Bonus functionality of the system. As the user clicks on the cluster, the topic modelling using LDA is performed and the topics with the corresponding tweets are visualized as cards. On hovering on the node, other details like the main hashtags and the number of tweets in the cluster are displayed.

ics present in the tweets of the cluster, as depicted in figure 6. The third bonus feature that we implemented was that upon clicking on the nodes of the graph, the user can manually inspect the individual tweets of the cluster, as depicted in 6. All these features enable the user to perform detailed, fine-grained analysis of the data set that other tools like pyLDAvis [Sievert and Shirley, 2014] and pyVis [Institute, 2018] don't provide.

10 Conclusion and Future Work

Our project allows the users to interactively vary different parameters like edge threshold and distance threshold to see formation of different clusters of tweets and also distances i.e relationships between them. By using the tool, users can find the optimal values of distance and edge threshold that suits their tasks. The project helps in finding topics within the clusters formed and also the tweets with in clusters easily. The users with minimal knowledge about topic modeling and natural language processing can use the tool to derive valuable insights from the tweets.

Our tool could be enhanced with more additional functionalities like summarizing the tweets that are present in the cluster. This would be an added advantage for the user by helping them to understand the raw text without actually reading the full set of tweets in the cluster. Another possibility is to implement more than one clustering algorithms, so that the user can choose between multiple Unsupervised Clustering algorithms and compare the results to better understand how each clustering technique works.The tool could be further improved by allowing the users to upload the tweets on which they want the analysis and visualization to be performed.

References

[Blei, 2012] Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Dash, 2020] Dash (2020). Dash Documentation & User Guide — Plotly. https://dash.plotly.com/.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Hagberg et al., 2008] Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.

[Institute, 2018] Institute, W. H. (2018). Pyvis: Interactive network visualizations. https://pyvis.readthedocs.io/en/latest/.

[Kaggle, 2020a] Kaggle (2020a). 6.5M Coronavirus tweets (8 March - 24 April 2020). https://kaggle.com/datarefiner/65m-coronavirus-tweets-8-march-24-april-2020.

[Kaggle, 2020b] Kaggle (2020b). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

[Řehůřek and Sojka, 2010] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Sanh et al., 2019] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*.

[Sievert and Shirley, 2014] Sievert, C. and Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA. Association for Computational Linguistics.

[Wang et al., 2018] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint 1804.07461.

[Zipf, 1949] Zipf, G. K. (1949). Human behavior and the principle of least effort. *Journal of Clinical Psychology*, 6(3):306–306.