Université de Lille

Faculté d'Ingénierie et Management de la Santé (ILIS)

Master Data Science pour la Santé

MÉMOIRE DE FIN D'ETUDES DE LA 2ème ANNÉE DE MASTER

A Big Data Framework for Sentiment Analysis in Healthcare: the case of Covid-19

Par Ulrich TCHUENKAM

Sous la direction de Wajdi DHIFLI

2019/2020

Master 2 Data Science pour la Santé

Composition des membres du jury :

Président de jury : Benjamin GUINHOUYA, Enseignant-Chercheur

2ème membre du jury : Djamel ZITOUNI, Enseignant-Chercheur

3ème membre de jury : Mohamed LEMDANI, Enseignant-Chercheur

Date : 20 Octobre 2020

Faculté d'Ingénierie et Management de la Santé - ILIS

42 rue Ambroise Paré

F-59120 LOOS

Ulrich TCHUENKAM

UN CADRE BIG DATA POUR L'ANALYSE DES SENTIMENTS DANS LE SECTEUR DE LA SANTÉ: LE CAS DU COVID-19

Résumé :
D'énormes quantités de données textuelles sont produites, en particulier sur les plateformes sociales, à un rythme extrêmement rapide. Par conséquence, des technologies Big Data appropriées émergent ou évoluent constamment pour stocker, traiter et extraire des informations à partir de ces données textuelles massives. Ce projet vise à développer un cadre qui exploite les technologies de Big Data pour fournir une interface interactive pour effectuer à la fois une analyse de sentiments et une modélisation de sujets sur des données textuelles.
Tout d'abord, nous avons développé un cadre Big Data qui effectue une analyse des sentiments et une modélisation de sujets sur un ensemble de données textuelles fourni par l'utilisateur.
Deuxièmement, Twitter est une excellente source de données textuelles sur des sujets d'actualité dans divers domaines, notamment la santé. Nous avons extrait des tweets centrés sur la pandémie de Covid-19 et analysé ces tweets comme une étude de cas expérimentale pour notre cadre.

Mot-clés : Covid-19, NLP, analyse des sentiments, modélisation des sujets, Big data

A BIG DATA FRAMEWORK FOR SENTIMENT ANALYSIS IN HEALTHCARE: THE CASE OF COVID-19

Abstract :
Huge amounts of textual data are produced, especially in social platforms, at an extremely fast pace. Consequently, appropriate big data technologies are constantly emerging or evolving to store, process and extract insights from these massive textual data. This project aims at developing a framework that leverages big data technologies to provide an interactive user interface to perform both sentiment analysis and topics modeling on textual data.
Firstly, we developed a big data framework that performs sentiments analysis and topics modeling on a user-provided textual dataset.
Secondly, Twitter is a great source of textual data about hot topics in a variety of domains including healthcare. We extracted tweets centered around the Covid-19 pandemic and analysed these tweets as an experimental case study for our framework.

Keywords: Covid-19, NLP, sentiment analysis, topic modeling, Big data

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The fast multiplying quantity of data on the internet and in the healthcare industry has made inevitable the adoption of big data technologies to improve the quality of healthcare systems. Many researchers have raised attention to the potential big data approaches are offering in health analytics. Recently, the healthcare industry has witnessed a significant increase in the number of data sources as a result of the widespread use of mobile and wearable sensor technologies, which has overloaded the healthcare industry with a huge amount of data. It therefore becomes challenging to perform healthcare data analysis based on traditional methods which are unfit to handle the high volume of diversified medical data. The incorporation of big data technologies in healthcare analytics presents a valuable asset to boost the performance of medical systems.

(Chen et al., 2014) refers to big data as large datasets that combine the following characteristics: **volume** which refers to high amounts of data, **velocity** which means that the data is generated at a rapid pace, **variety** which emphasizes that data comes under different formats and, finally, **veracity** with means that data originates from trustable sources.

Some other properties of big data have been identified: the **variability** which indicates variations that occur in the data flow rates, complexity which arises from the fact that big data is often produced through a set of sources.

Additionally, (Oracle, 2020) mentioned value as a key attribute of big data. According to Oracle, big data has a low value density meaning that raw data has a low value compared to its high volume. However, analysis of important volumes of data may lead to obtaining a high value.

(Liu et al., 2015) proposed a prototype of Healthcare Big Data Processing System based on (*Apache Spark*, 2012) to analyze the high amount of data generated by healthcare big data process systems. This solution is based on spark which is very promising since it handles batch computing, stream computing, and ad hoc query.

(Bochicchio et al., 2016) presented a big data healthcare analytics framework for supporting multidimensional mining over big healthcare data. The objective of this framework is analyzing the huge volume of data by applying data mining methods.

(El aboudi & Benhlima, 2018) demonstrated the potential of using big data technologies in the healthcare industry to find useful information in highly valuable data by proposing an extensible big data architecture for healthcare applications. This architecture is based on both stream and batch computing to make accurate predictions on patient health conditions. Using (*Apache Spark*, 2012) and (*MongoDB*, 2007) tools they demonstrated the potential of their architecture in generating real time alerts in a healthcare system.

These past works suggest that a good number of analytical studies are leveraging big data technologies for their efficiency. Analytical studies performed in the health sector can generally be grouped into: **descriptive analytics** which consist of describing current situations and reporting them, **diagnostic analysis** which aims to explain why certain events occurred and what factors triggered them, and **predictive analytics** which reflects the ability to predict future events; it also helps in identifying trends and determining probabilities of uncertain outcomes. Sentiment Analysis(SA) and Topic Modeling(TM) which are part of our interest in this work both fall under the category of predictive analytics studies.

Sentiment Analysis is concerned with the investigation of opinions, thoughts and feelings. It is used as a tool to understand Natural Language Processing (NLP). It aims to determine the thoughts of the speaker or writer regarding a specific subject or topic or simply to identify the overall polarity of a document (classification) (Li & Dash, 2010). In other words, it extracts and retrieves information from unstructured raw data, which are usually presented in the form of judgement or evaluation and reflects any kind of emotion. Much can be gained by extracting sentiments from healthcare information, including views from beneficiaries such as patients.

Topic modeling is a popular statistical tool for extracting latent variables from large datasets(Blei, 2012). One of the most critical goals of data analytics is determining the characteristics that data points share. In text analysis, this often means determining what events or concepts a document is discussing. This information is clear to a human reading a document, but a program is given only the text as it is written, not the subject matter of each document. In order to accomplish this task in a program, data scientists usually resort to topic modeling.

The contribution of this work is the suggestion of an extensible big data framework that could be used in the healthcare industry to perform sentiment analysis and topic modeling. The framework consists of several components capable of mining, processing, storing and analysing high amounts of data in a batch mode. Our work uses tweets collected on Covid-19 to demonstrate the potential of using big data technologies to analyse sentiments and model topics on trending subjects of the health sector.

2. METHOD

We are developing a framework that has the advantage to be generic and can deal with textual data from any source. In this study, we propose a framework that aims at handling big textual data originating from heterogeneous sources in various formats like Comma Separated Values (.csv), Excel Spreadsheets (.xlsx), text files (.txt) or JSON (.json) documents. Moving from bottom to top, the scenario in Figure 1 illustrates the data management in the framework.

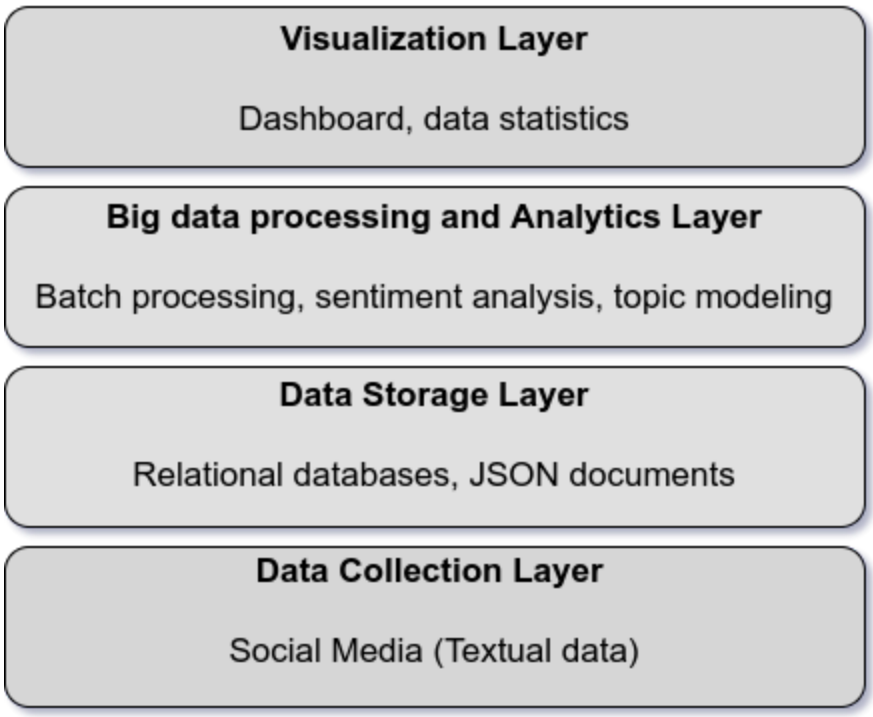


Figure 1. The layer architecture.

2.1. Data collection Layer

This component of the framework consists of the various data sources to which the framework can connect and retrieve data. The sources of data here could be Electronic Healthcare Records, social media, forums, and medical reports. The acquired data can undergo some basic transformations or they could directly be saved as raw data in the database configured in the data storage layer. In the proposed system, the textual data collected from heterogeneous sources can be classified into structured data such as EHR, or semistructured data such as XML, CSV and JSON documents.

2.2. Data Storage Layer

Data Storage is one of the most challenging tasks in a big data framework, especially in the case of healthcare systems which involve large amounts of data. Therefore, traditional data analysis is unfit to manage those systems. This component could leverage one or a combination of existing big data storage systems may be HDFS, NoSQL such as MongoDB, hive and elasticsearchSQL databases, or a combination of all of them. This allows it to beTherefore, it is more scalable and ensures high storage capabilities. In the proposed system, the textual data collected from heterogeneous sources can be classified into structured data such as EHR, or semistructured data such as XML, CSV and JSON documents. These data will be stored into appropriate raw data stores in the target databases.

2.3. Big Data Processing and Analytics Layer

Batch computing is performed on extracted data from the prepared data store through different phases.

2.3.1. Data Preparation

Generally, the acquired big data contains more variables and entities than required for a particular study. The data preparation component should assure the preparation of the data for a particular study. The proposed data preparation component consists of filtering and cleaning the dataset.

2.3.1.1. Data filtering and wrangling

The filtering component would ensure that we filter the data and retain just the variables and entities of interest. Meanwhile the data wrangling component will ensure that the data follows a set of norms based on the corpus of choice. Given that we are working with textual data, the component ensures that we work on a defined corpus. Text clean up and normalization mainly involve tasks like stemming and lemmatization where words are brought to their root-words, special symbols, punctuation marks and stopwords are removed from the text. This component allows the use of a personalized corpus and list of stopwords.

2.3.2. Sentiment Analysis

This component of the big data processing layer should be responsible for the analytics of the feelings hidden in the textual data. Feelings will be categorized in terms of polarity and subjectivity. In terms of polarity, a text document could be placed in one of the three labels: neutral, positive or negative. And in terms of subjectivity, a text document could be placed in one of the three labels: neutral, subjective or objective.

2.3.3. Topic modeling

The topic modeling component of the big data processing layer should be responsible for the analysis of the text and returns a number of key subjects or topics hidden in the text. Parameters of this component could be customized, for instance, in terms of the number of topics to model and the probability of retention of a topic.

2.4. Visualization Layer.

The visualization layer provides the interface for the user to interact with the batch processing and analytic layers. The output of the analytics are equally presented to the user by this layer. Presented outputs include: general statistics about the data input at the data layer, the output of the sentiment analysis, and the output of the topic modeling.

Figure 2 displays the details of the proposed big data framework for sentiment analysis in healthcare. Note that the arrows in the figure are depicting the direction of information and data flow between the framework components.

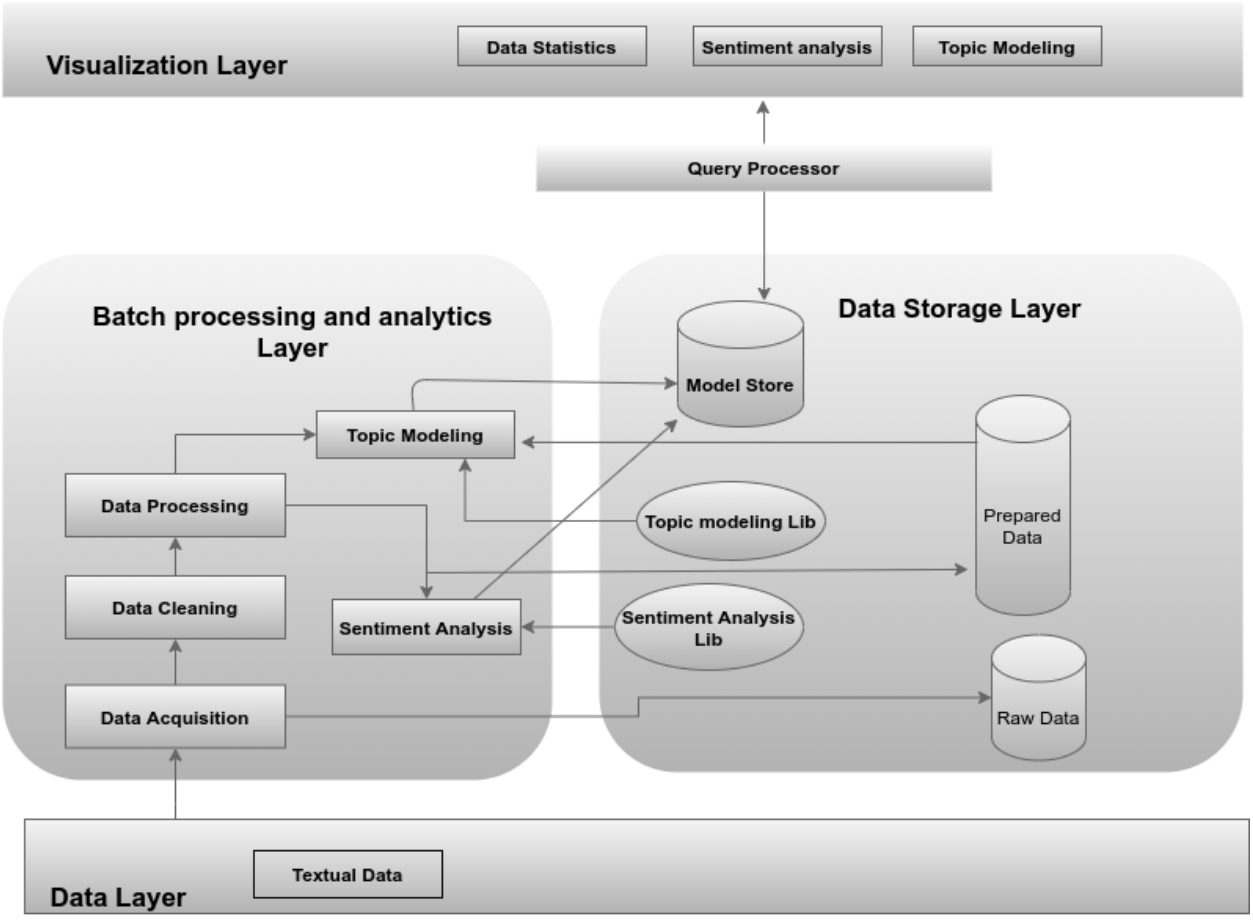


Figure 2. Big data framework for topic modeling and sentiment analysis in healthcare

3. CASE STUDY ON COVID-19

In this project, we aim to use the architecture described above to analyse sentiments and model topics from textual data collected from Twitter. The outbreak of the pandemic disease covid-19 happened while we were developing this project and we immediately thought of implementing a prototype based on tweets centered around the covid-19. Following the proposed architecture, the data layer in this prototype is a social media; specifically Twitter. Every code implementation was done using Python programming language.

3.1. Data acquisition and storage

Twitter has put at the disposal of researchers an API for sending and reading tweets programmatically. Part of the dataset used in this project was extracted from Twitter using their API. However, the Twitter free API imposes a limit on the number of extracted tweets. Thus, we also use a public dataset made available by (Md. Yasin & Sanjay, 2020) in the form of daily csv files during the covid-19 outbreak. A python script has been implemented to automatically connect to Twitter and download the required tweets. The downloaded tweets were automatically stored in a PostgreSQL database and equally indexed into ElasticSearch. For this analysis, we only extracted tweets that were written in the English language. This is important for the consistency of the sentiment analysis and topic modeling results.

3.2. Dataset

The dataset we are using in this prototype consists of 633549 tweets written in english and centered around the subject corona. The tweets were collected for the first 2 weeks of April 2020 (2020-01-04 to 2020-14-04). The dataset contains three variables: one which contains the texted tweet, a second variable which stores the date the tweet was written, and lastly a third variable which contains the state of residence of the user who authored the tweet.

3.3. Data wrangling

The acquired raw data was updated with a cleaned version of itself. Data wrangling consisted of the following: the removal of non-alphabetic characters from the text, removal of punctuation marks, removal of emoticons, and removal of English stopwords.

3.4. Data processing

Prior to topic modeling, we have to prepare the dataset to ensure that our data is less noisy. The processing consisted of the following operations: tokenization which consists of splitting a text into a list of its its words, removal of stopwords, converting tokens to lowercase characters, converting tokens into a set of bigrams (pairwise combination of the tokens), stemming and lemmatization which consist of reducing every single word to its corresponding root-word.

3.5. Topic Modeling

Topic modeling is a popular statistical tool for extracting latent variables from large datasets (Blei, 2012)**.** Using topic modeling, we are able to infer new and unobserved variables from previously observed variables. Examples of applications of topic modeling include: structuring databases of journals and articles into groups based on similar focus (Blei, 2012), grouping social media users by similar post content (Hong & Davison, 2010), and grouping genomic data by similar sequence structure (Liu et al., 2016). Our prototype implements the algorithm called Latent Dirichlet Allocation (LDA) which is one of the most frequently used algorithms for topic modeling. In LDA, each document (tweet) could be viewed as a mixture of various topics where each document is considered to have a set of topics that are assigned to it via LDA. The python library called Gensim (*Gensim*, 2010) was used to model topics. Gensim is one of the most frequently used packages for this task and it supports parallel and distributed computing which are paramount for the execution of a big data framework. In this prototype, we are using Gensim multicore implementation of the LDA algorithm. Given that LDA exploits the frequency of occurrence of the tokens in the corpus, we transformed the corpus using a technique called Term Frequency - Inverse Document Frequency (TF-IDF) to limit the possible bias related to possible outliers. TF-IDF quantifies documents in the corpus by regularizing the frequency of a word in a document and the frequency of this word in the whole corpus. TF-IDF computes a score to each word which represents both the importance of the word in a document (local) and the importance of the word in the whole corpus (global). This technique automatically limits the effects of outliers on the resulting model.

TF, IDF, and TF-IDF can be formulated as follows:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

TF-IDF(t) = TF(t) x IDF(t)

3.6. Sentiment Analysis

Sentiment analysis is a classification task which aims to determine the thoughts of the speaker or writer regarding a specific subject or topic or simply to identify the overall polarity of a document (Li & Dash, 2010). (Kent et al., 2015) found that up to 40% of healthcare tweets contain some form of sentiment. (Greaves et al., 2013), observed that accurately measuring the sentiment of a healthcare tweet represents an opportunity for understanding both the patient's and healthcare professional's opinion on a health subject. For this prototype, we used the python library called Textblob (*TextBlob*, 2013) to extract the sentiment and classify a tweet in terms of polarity and subjectivity. TextBlob provides 2 types of sentiment analyzers: NaiveBayes analyzer and Pattern based analyzer. The Pattern analyzer returns sentiments in terms of polarity and subjectivity meanwhile the NaiveBayes analyzer returns sentiment just in terms of polarity. For this prototype, we used the pattern analyzer. In terms of polarity, a tweet is classified as either positive, negative or neutral and in terms of subjectivity, a tweet is either objective, neutral or subjective.

4. RESULTS

The product of this project is a web application developed with the python package called (*Dash open source*, 2015) which runs on (*Flask*, 2010) as the server. I will describe the results based on the various compartments of the web application.

4.1. Data Summary

The summary tab of the application loads a default dataset and equally allows the user to upload his own dataset as can be seen in Figure 3.

Figure 3. Interface for dataset upload.

Upon loading the dataset, the batch processing layer of the framework wrangles and prepares the dataset for the subsequent analysis. The batch processing layer then runs some statistics on the cleaned dataset and outputs the visualisations shown in Table 1 and Figure 4 below. Table 1 displays the number of tweets contained in the dataset and the size of the cleaned dataset in Megabytes while Figure 4 displays the daily variation in the number of tweets. We had the most tweets on April 13 and the least tweets on April 2. The number of tweets kept increasing from April 4th to April 10. Between April 9th and 14th, the number of tweets had never gone below 23,000.

Data Summary	
Total Number of Tweets	633549
Size of dataframe in MB	307.85

Table 1. Number of tweets in the dataset and size of dataset in megabytes.

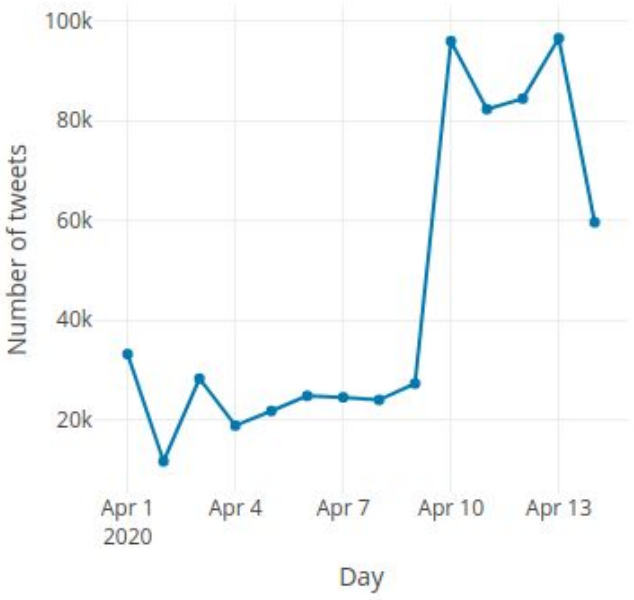


Figure 4. Number of tweets per day.

4.2. Topic Modeling

The web interface provides a tab to define the hyperparameters of the topic modeler before launching it. In Figure 5, a user has the possibility to define the hyperparameters of the model: the number of topics, the document-topic density (alpha), the topic-word density (beta), and the minimum probability for an accepted topic. The document-topic density controls the number of topics to which a document will be assigned. The higher is the document-topic density, the higher is the number of topics to which a document will be assigned and vise versa. The topic-word density controls the number of words assigned to a topic. The higher is the topic-word density, the higher is the number of words that each topic will contain. By varying the hyperparameters of the model we will measure the perplexity and coherence scores of the model for evaluation of the model's performance. The lower the perplexity the better the model and the higher the coherence score the stronger the model. Running the simple model will produce an interactive bubble plot as shown in Figure 6 while running the advanced model produces the interactive visualisations shown in Figure 7.

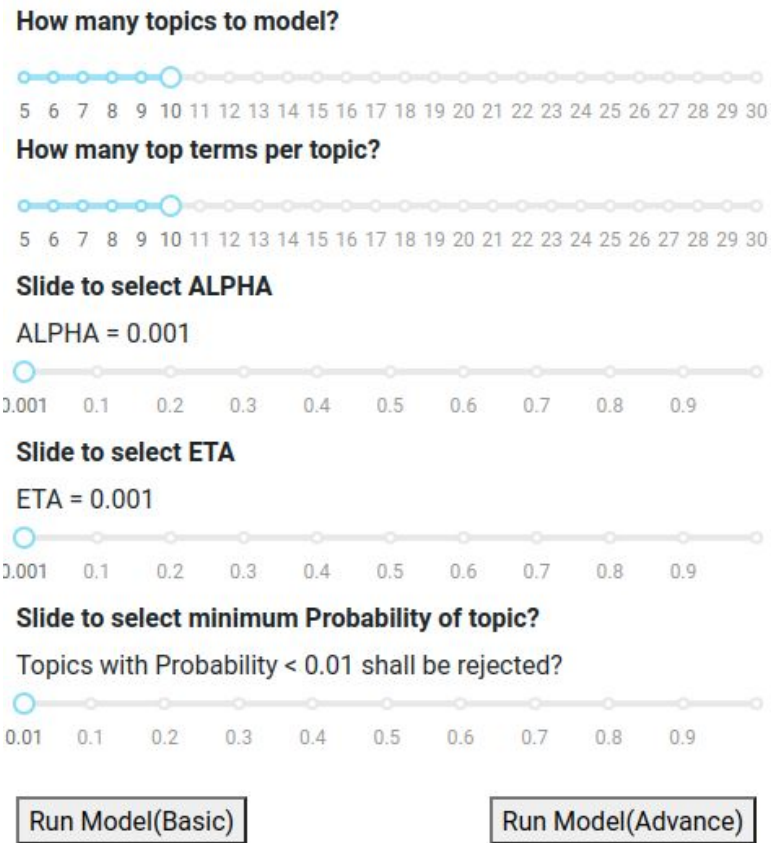


Figure 5. Interface to define the hyperparameters of the topic modeler.

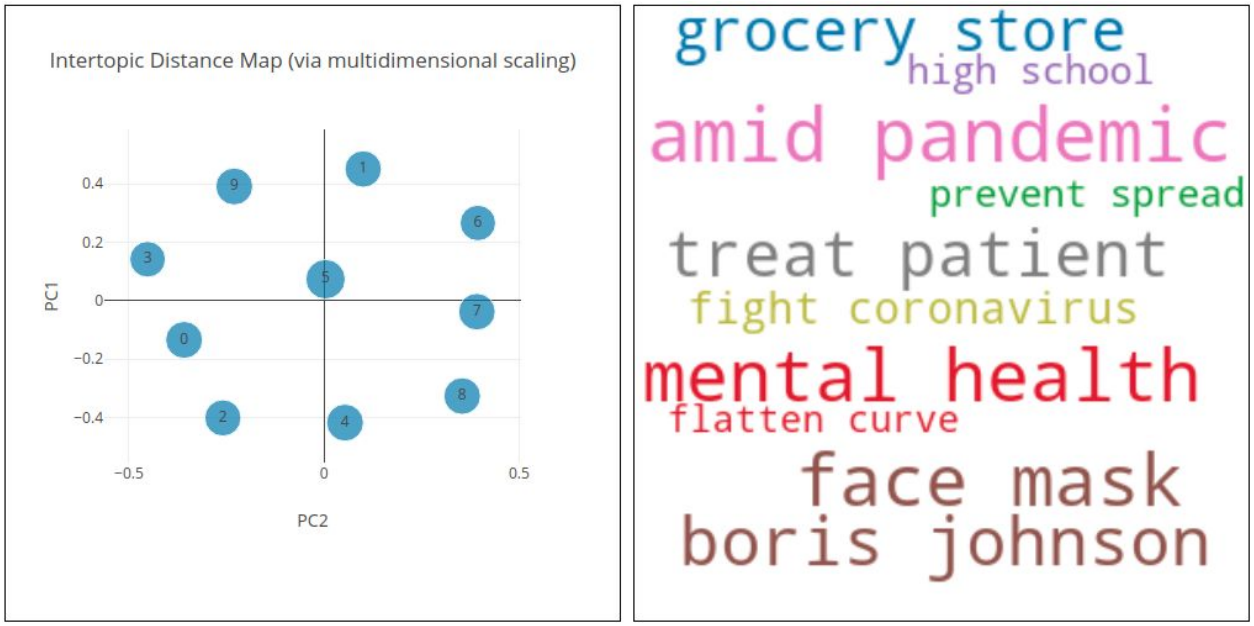


Figure 6. Results of running the simple topic modeling option.

The bubbles in Figure 6 represent the topics modeled and hovering the mouse over each topic will output its corresponding word cloud.

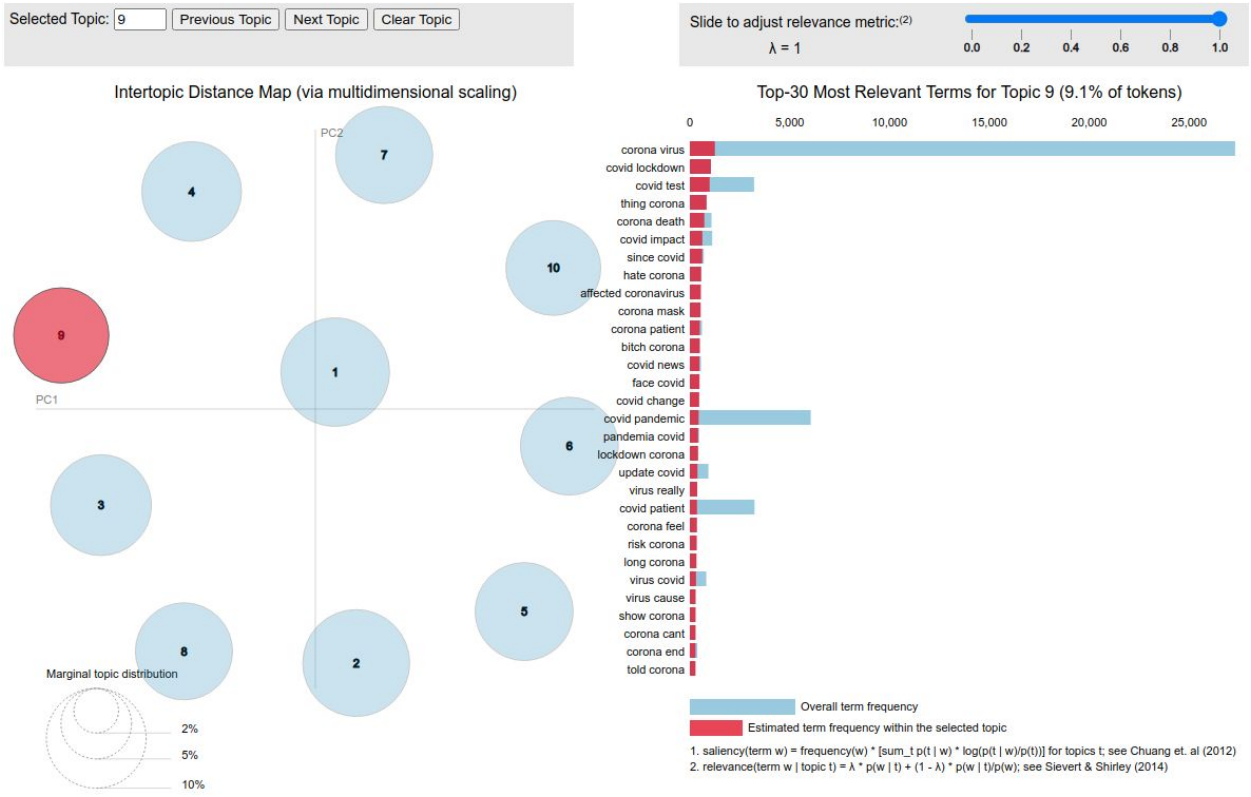


Figure 7. Results of running the advanced topic modeling option.

Similarly to Figure 6, the bubbles in the visualization in Figure 7 represent the modeled topics and hovering over each topic outputs the top 30 keywords associated with this topic on a bar chart. The blue bars display the frequency of the keyword in the overall corpus while the red bars display the estimated frequency of the keyword in the selected topic.

The topic modeler equally gives the possibility to the user to rename the discovered topics as shown in Figure 8**.** This eases the identification of the topics by the user in the other interfaces of the application.

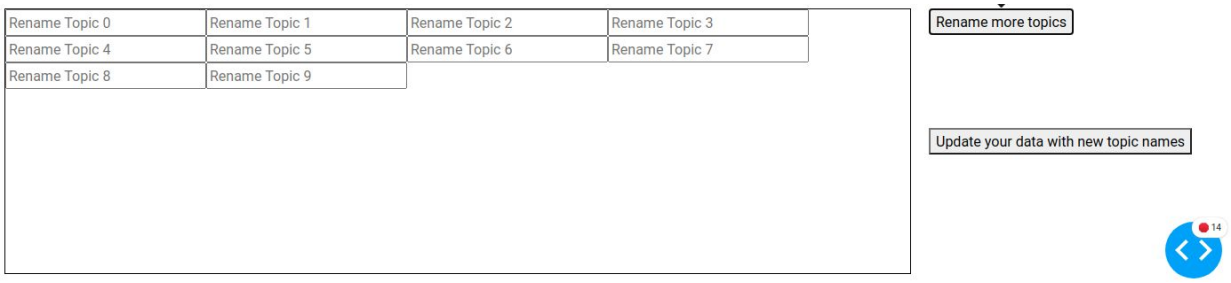


Figure 8. Renaming the discovered topics by the topic modeler

4.3. Sentiment Analysis

Running the topic modeler as described in the previous section equally runs the sentiment analysis model on the tweets and the results of the sentiment analysis is available on a separate tab of the application. Two flavours exist for the sentiment analysis: one based on all the tweets of the dataset and a second one based on the topics created in the topic modeling.

4.3.1. Sentiment analysis on tweets

Figure 9 and Figure 10 display bar plot visualizations of the results of the overall polarity and subjectivity contained in the dataset. In terms of polarity, it can be seen that most users are neutral to the subject. Surprisingly, it can also be seen that the fraction of positive sentiments (tweets) is higher than that of negative sentiments by almost two orders of magnitude. In terms of subjectivity, it can be seen that most users are objective.

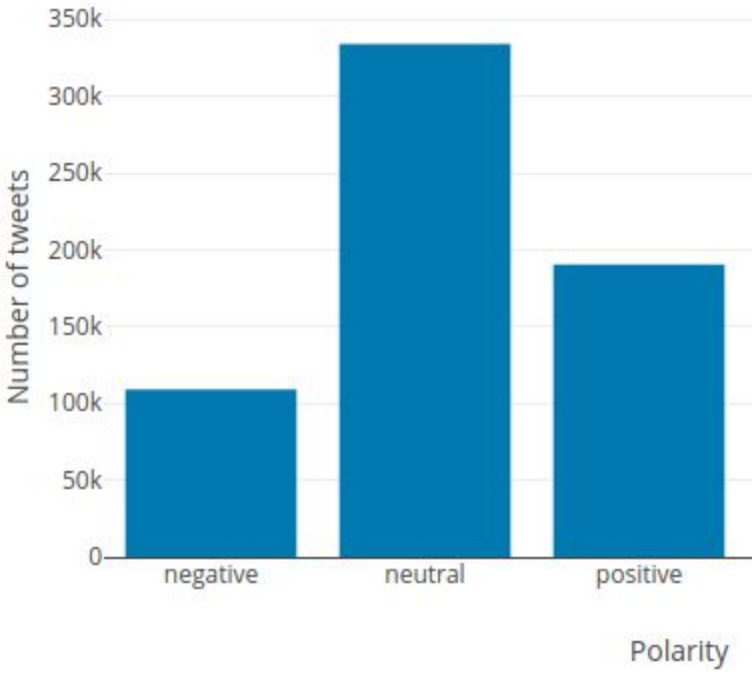


Figure 9.Sentiment analysis (polarity) on tweets.

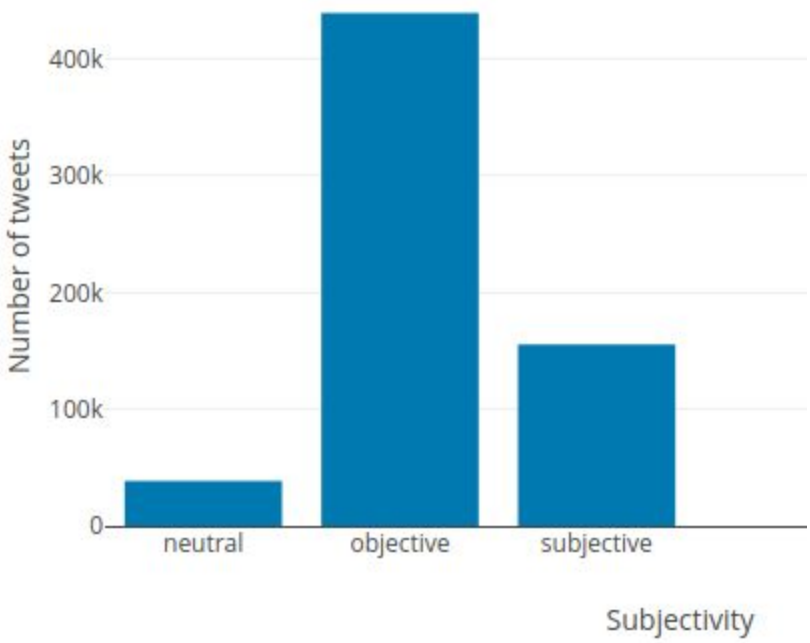


Figure 10. Sentiment analysis (subjectivity) on tweets.

4.3.2. Sentiment analysis based on topics

Another result of the sentiment analysis was obtained by aggregating the sentiments based on the topics and computing the percentage of the different sentiments in each topic. Figure 11 shows a pie visualization for one of the topics created earlier (topic 7).

In terms of polarity, It can be seen that 53.8% of the tweets in, for instance, topic 7 are neutral while 29.5% are positive and only 16.7% are negative.

In terms of subjectivity, it can be seen that 71.5% are objective while 21.8% are subjective and only 6.6% are neutral.

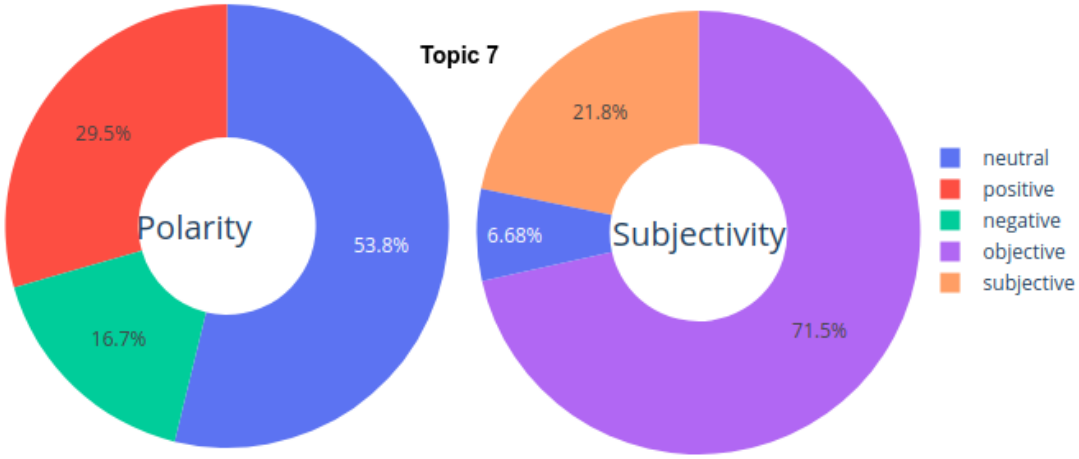


Figure 11. Sentiment analysis based on topics.

5. DISCUSSION

In this study, we have developed a framework that leverages big data technologies to highlight the key topics discussed in a set of textual data and classify the sentiments contained in this set of data. As an experimental case, we extracted a set of tweets centered around the Covid-19 and analysed it using the framework.

The particularity of this work is the fact that it leverages big data tools which support parallel and distributed computing thus making the framework scalable and fit to support huge volumes of datasets. In addition to scalability, the framework doesn't operate only on tweets but rather accepts any set of textual data since it abstracts the data collection layer which could be adapted to the source of data (tweeter, Facebook posts, news articles, RSS fluxes, scraping modules, etc.).

Though the Twitter API provides a great way to retrieve tweets, there are a number of limitations associated with the API. Indeed, we can't access tweets more than 7 days older which is a bottleneck for anyone looking for older tweets. Other APIs like (Ahmet & Lasse, 2017) and (Dmitry, 2019) give access to older tweets but only get a random number of them .

Though our work provides a general framework for sentiment analysis and topic modeling, some additional features could be implemented to boost the framework. One such feature could involve the implementation and evaluation of other topic modeling algorithms since our framework currently implements only LDA for topic modeling.

Another possible additional feature could consist of leveraging Spark Streaming API to add support for the automatic extraction and real-time analysis of tweets without having to always download the tweets before analysing later. With such a feature, a user could have an overview of what is being discussed on Twitter at a particular point of time.

One could also think of incorporating a user mapping feature into the framework to create potential networks between users as this can provide a floor for the identification of influential users. Identifying influential users can give way to the analysis of the potential effect that some users have over the others or predict the nature of the relationship betweens users in a network.

The support for several corpus can be something very interesting to have in the framework as users might have text in languages other than English. Compared to the other languages , English has the most developed corpus and recently multiple communities are working hard to have efficient corpus for the other languages like French.

6. CONCLUSION

There are abundant sources of huge textual data around us and Twitter is one such great source. Twitter provides an environment to share ideas and discuss trending

topics. By leveraging big data technologies (spark and elasticsearch) and the abundant textual data from Twitter, we presented a prototype of our proposed big data supporting framework for extracting and highlighting the key subjects contained in a set of tweets in addition to highlighting how the user feels about a particular subject.

7. REFERENCES

Ahmet, T., & Lasse, S. (2017, Jan 10). *twitterscrape*. GitHub. Retrieved Sep 25, 2020, from https://github.com/taspinar/twitterscraper

Apache Spark. (2012). Apache Spark - Unified Analytics Engine for Big data. Retrieved Sep, 2020, from https://spark.apache.org/

Blei, M. B. (2012). Probabilistic topic models. In *Communications of the Association of Advanced Machinery* (4th ed., Vol. 55, p.77). DOI:10.1145/2107736.2107741

Bochicchio, M., Cuzzocrea, A., & Vaira, L. (2016, December). A big data analytics framework for supporting multidimensional mining over big healthcare data. *in Proceedings of the 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016,*, 508–513.

Chen, M., Mao, S., & Liu, Y. (2014). “Big data: A survey,”. In *Mobile Networks and Applications* (2nd ed., Vol. 19, pp. 171–209).

Dash open source. (2015). Dash overview. Retrieved Sep, 2020, from https://plotly.com/dash/

Dmitry, M. (2019, Nov 27). *GetOldTweets3*. PyPi. https://pypi.org/project/GetOldTweets3/

Ela, A., & Özgür, K. (2020). Topic Modeling of Twitter Data via RapidMiner. *Bilgi Yönetimi*, *3*(1), 1-10. doi:10.33721/by.641878

El aboudi, N., & Benhlima, L. (2018). Big Data Management for Healthcare Systems: Architecture, Requirements, and Implementation. *Advances in Bioinformatics*, 1–10. doi:10.1155/2018/4059018

Flask. (2010). Welcome to Flask. Retrieved Sep, 2020, from https://flask.palletsprojects.com/en/1.1.x/

Greaves, F., Ramirez-Cano, D., Millett, C., Darzi, A., & Donaldson, L. (2013). Use of sentiment analysis for capturing patient experience from free-text comments posted online. In *Journal of Medical Internet Research* (15th ed., Vol. 11, pp. e239-e251). doi: 10.2196/jmir.2721

Hong, L., & Davison, D. B. (2010). Empirical study of topic modeling on Twitter. *in Proceedings of the First Workshop on Social Media Analytics - SOMA ’10*, 80–88.

Kent, E., Gaysynsky, A., Galica, K., Rinker, R., Graff, K., & Chou, S. W.Y. (2015). Obesity is the new major cause of cancer: connections between obesity and cancer on facebook and twitter. *Journal of Cancer Education.*, *31*(3), 453-459. doi: 10.1007/s13187-015-0824-1

Li, N., & Dash, W. D. (2010). Using text mining and sentiment analysis for online forums hotspot detection and forecast. *48*(2), pp. 354 – 368. http://www.sciencedirect.com/science/article/pii/S0167923609002097

Liu, L., Tang, L., Dong, W., Yao, S., & Zhou, W. (2016, Sep). “An overview of topic modeling and its current applications in bioinformatics,”. *Springerplus*, *5*(1).

Liu, W., Li, Q., Cai, Y., Li, Y., & Li, X. (2015). A prototype of a healthcare big data processing system based on Spark. *8th International Conference on Biomedical Engineering and Informatics (BMEI)*. doi:10.1109/bmei.2015.7401559

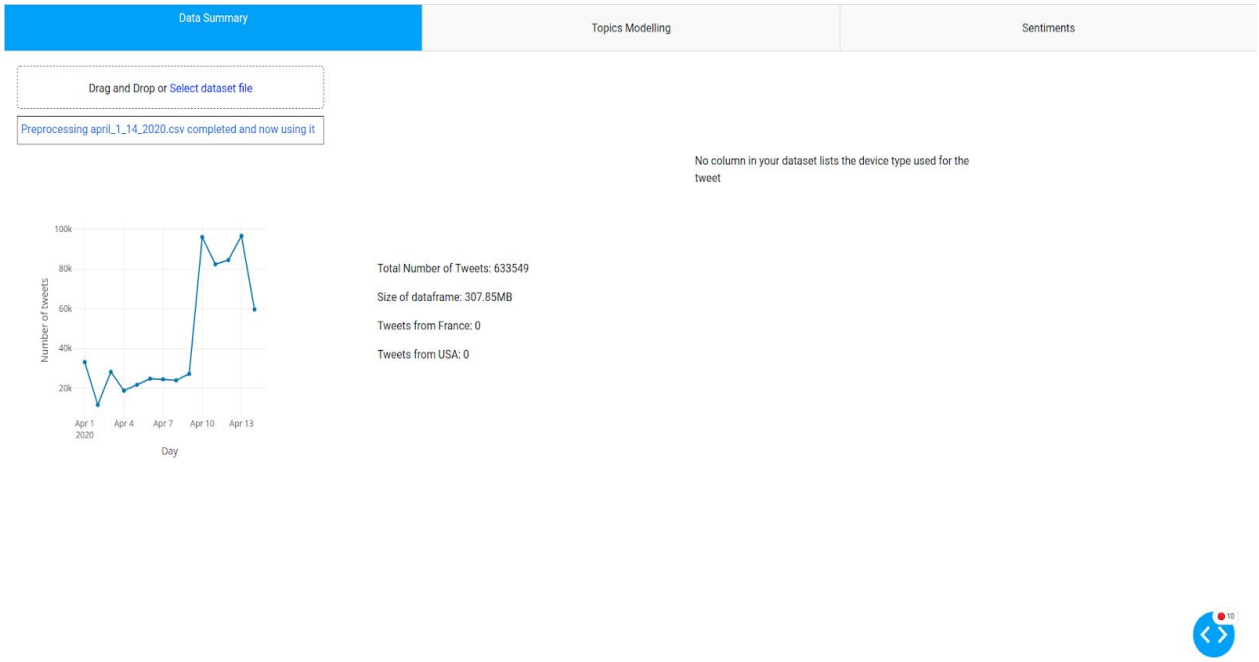
Md. Yasin, K., & Sanjay, M. (2020). *CoronaVis: A Real-time COVID-19 Tweets Analyzer*. GitHub. https://github.com/mykabir/COVID19

MongoDB. (2007). The most popular database for modern apps | MongoDB. Retrieved Dec, 2020, from https://www.mongodb.com/

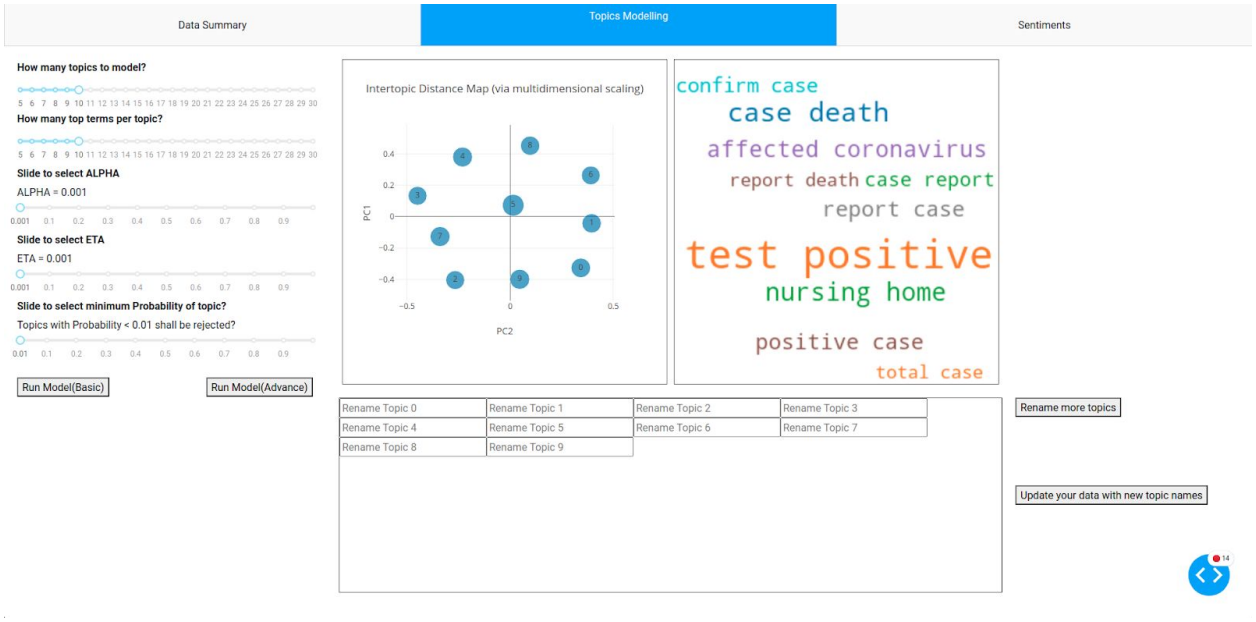
Oracle. (2020). *Oracle*. What is Big Data? Retrieved Oct 2, 2020, from https://www.oracle.com/big-data/what-is-big-data.html

8. ANNEX

ANNEX 1: THE WELCOME AND DATA SUMMARY TAB OF THE PROTOTYPE



ANNEX 2 : THE TAB TO CONFIGURE AND LAUNCH THE TOPIC MODELER AND SENTIMENT ANALYZER.



ANNEX 3 : THE TAB TO VISUALIZE AND INTERACT WITH THE RESULTS OF THE SENTIMENT ANALYSIS.