Data Mining Project's Report

Aurora Maria Tumminello
University of Trento
1st year, Data Science, Trento, Italy
aurora.tumminello@studenti.unitn.it

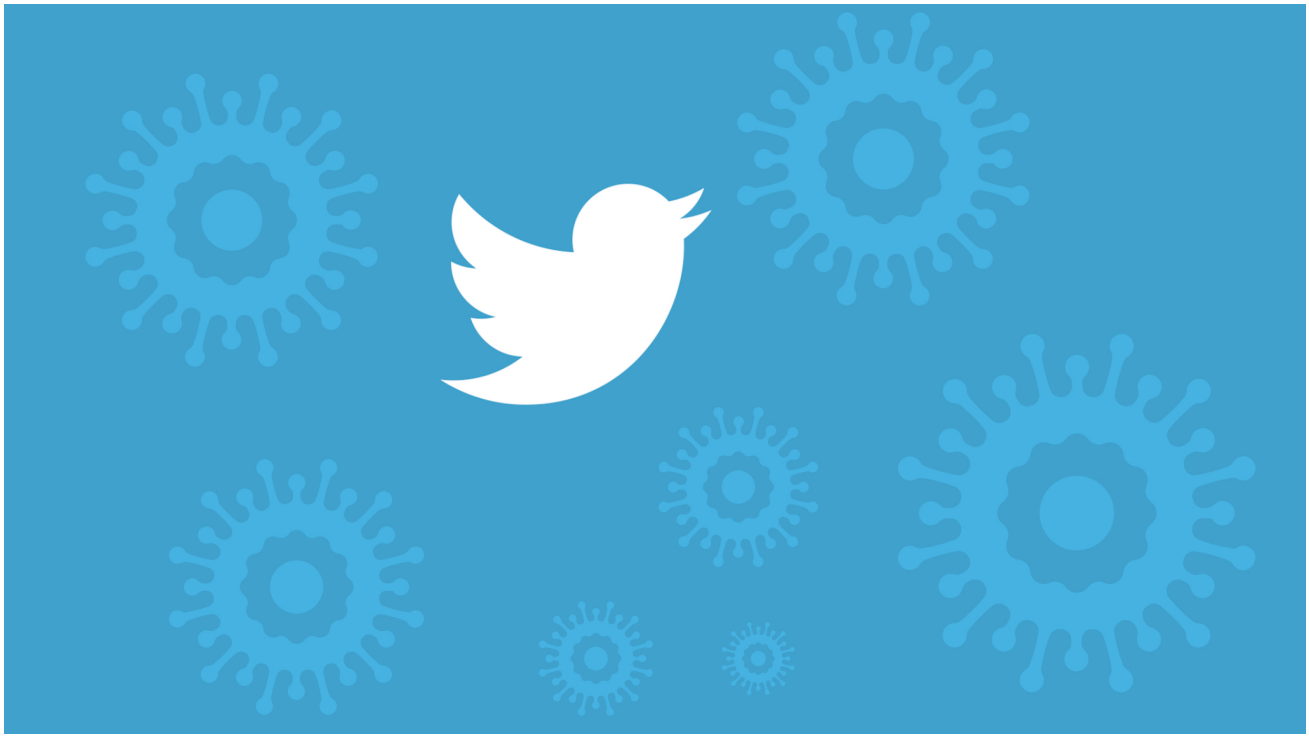

Figure 1: Twitter and COVID19

ABSTRACT

This report provides a description of the work related to the data mining project of the academic year 2020/2021. An introduction and motivation will be provided at the start, accompanied by some references to related work from a theoretical perspective. Furthermore, the problem statement and a detailed explanation of the solution proposed will be presented. In conclusion, the report will contain some experimental evaluation.

1 INTRODUCTION AND MOTIVATION

The project presented focuses on Twitter data in a specific time period and it aims to identify popular topics contained in tweets. A topic can be considered as a set of terms that regularly can be found together in tweets and that it is popular at a certain time.

Before talking deeply about the project, some motivations are provided. First of all, we will focus on **Twitter data**, because they provide a clear overview of events and news in the political, cultural and sociological worldwide sphere. Secondly, **tweets** are short texts sent at specific times by users all over the world, thus people have to be more directed. In order to do so, they commonly use hashtags, i.e. words preceded by the `#` character, to spread the word and communicate directly the message with few words. It is interesting to notice that redundancy and ambiguity of natural language are diminished thanks **hashtags**: they are easy to read, to follow and they communicate directly an idea. Moreover, they become word of mouth and they are frequently used to communicate something that maybe has multiple meanings or way to be told. In addition, Twitter data, contrary to other social networks, are open and downloadable through its API, despite there is a download limit.

The usefulness of this project resides in understanding what is happening in the world, taking into account time and place (when location information is available). Furthermore, to reach this goal, the project aims seeing how words are correlated by looking at how many times they appear together and if they can be considered as a frequent itemset, i.e. as a topic.

A relevant point to take into consideration is the **timing** factor, because it bestows a contextual background to topics discovered. If data contextualization was not acknowledged, we would lose all the necessary background for accurate data analysis and interpretation. Therefore, the project presented will look at consistent sets of terms that are frequently used throughout time, i.e. popular topics.

2 RELATED WORK

This section aims to provide some useful theoretical works that could help to understand the following steps of this report.

date	text
2020-07-25	['smell', 'scent', 'hand', 'sanit', 'today', 'someon', 'past', 'would', 'think', 'intox']
2020-07-25	['hey', 'yanke', 'yankeespr', 'mlb', 'made', 'sens', 'player', 'pay', 'respect']
2020-07-25	['dian', 'wdunlap', 'realdonaldtrump', 'trump', 'never', 'onc', 'claim', 'covid', 'hoax', 'claim']

Table 1: Example of input file

itemsets	n_items	date	support
[covid]	1	[2020-07-24, 2020-07-25, 2020-07-26, ...]	0.598301
[case]	1	[2020-07-24, 2020-07-25, 2020-07-26, ...]	0.091421
[coronavirus]	1	[2020-07-24, 2020-07-25, 2020-07-26, ...]	0.077630

Table 2: Output of the project presented

Before starting to discuss algorithms that could be useful, some text processing clarifications may be necessary. Given the disordered and redundant nature of natural language, data preprocessing is a necessary task in order to evaluate all texts equally and without special characters or distinctions for spaces or case sensitivity. Therefore, taken tweets texts, some steps need to be performed. In particular, we should recur to:

- **RegEx** (regular expression) can be used to identify textual patterns and modify tweets before working on data;
- **stopwords removal**: stopwords are very common words that could alter our analysis with useless topics. For instance, "*and*" could be used in all tweets and all popular words would be in the same set with it, despite it is not so meaningful;
- **stemming**, which is the process of reducing inflexion in words to their root forms. For instance, if we have "*plays*", "*playing*" and "*playful*", the stemmer will reduce all words to "*play*".

3 PROBLEM STATEMENT

The input of the main algorithm presented is a dataset with the date and the cleaned text corpus of tweets, therefore of dimension $N \times 2$, where N represents the number of rows. The `date` column should be a string in the format `YYYY-mm-dd`, while `text` contains tweet body, entire or truncated. If the corpus is truncated, there will be suspension points followed by the link to the referred tweet. The input file should have extension `.csv`. An example of the input file can be found in Table 1.

Notice that the solution displayed in this report accepts all dataset that respects those assumptions. Therefore, it can be applied to the **COVID dataset** used as suggested by the project description, as well as any other Twitter dataset.

The solution proposed will return as output a set of significant frequent itemsets discovered in the input dataset. The topics contained in the output will be ordered increasingly by the popularity of that topic, i.e. the more that topic is cited in the dataset, the highest position will cover in the results. More in detail, the output will contain: `itemsets`, a list of frequent words recurring together; `n_items` with the number of words in `itemsets`; `date`, which contains a list of all dates in which the itemset has been frequent; `support`, or the frequent itemset occurrence in proportion to the total number of observations in the dataset. Table 2 shows an example of the output described.

4 DATASET

As depicted in the problem statement, the input should detain only two columns: date and processed text of tweets. Despite that, the original dataset related to COVID19 disposes of the following columns: `user_name`, `user_location`, `user_description`, `user_created`, `user_followers`, `user_friends`, `user_favourites`, `user_verified`, `created_at`, `tweet`, `hashtags`, `source` and `is_retweet`. As can be noticed, the original dataset contained information about the user who published the tweet, such as username, description, number of followers and location. Related to the latter information (i.e. places where tweets' authors come from) are omitted since there are many missing values. Moreover, location is inserted by users and therefore there is no arbitrary way to indicate, for instance, New York (e.g. *'New York, New York', 'New York, NY, USA', 'Brooklyn, New York', 'New York, LA, Miami, Houston'*). Thus, this project will focus more on the timing instead of the place where topics come from.

In order to obtain the cleaned version of tweets, some computations are necessary, which are contained into `text_processing.py` file provided:

1. `import_dataset()`: select only the necessary columns from the original dataset (`created_at` and `tweet`), changing their names to 'date' and 'text'. Moreover, only year, month and day are considered, through 'date' column conversion to datetime.
2. `remove_links()`: removal of internal and final links from tweets (i.e. those words that start with http);
3. `remove_symbols()`: this function removes all symbols from tweets (i.e. *#, @, points, \n, \r, suspension points* cause by text truncation);
4. `only_text()`: the resulting text is divided into many strings, according to the space separator. All non-alphanumerical characters are erased, words are converted to lower case and all remaining symbols or numbers interword are deleted. Notice that numbers were erased only at this step because some words contain numbers in it, such as *covid19* or *covid-19*. In order to disambiguate terms like the previous two (e.g. *covid19, covid-19, covid*), numbers were not considered at all. In the end, empty strings are removed;
5. `remove_stopwords()`: taking into consideration only the English vocabulary, all words obtained from previous phases are stemmed to the root. Finally, stopwords are removed, in order not to consider obvious and frequently recurring words, such as *and*.

All these functions are merged inside `text_processing()`, which is called by `clean_tweets()` for being computed for each row in the dataset. The new dataframe is then saved in an external file as `input.csv`. Within the project, output files are saved in two formats: `csv`, since it is universally readable; `pickle` format, which allows maintaining columns types without any casting whenever there is the need to import the file.

In addition to what said about the COVID dataset, it is available at https://www.kaggle.com/gpreda/covid19-tweets and it refers to the time period from 2020-07-24 to 2020-08-30. The dataset contains 179,108 observations, with 112,733 unique words obtained after text processing. As can be seen in figure 2, where the 20 most recurring terms are shown, nearly all terms occur less

than 20K times, except for **case** and **covid**, which is the most occurring word in all dataset. Therefore, covid will be probably present in a large number of frequent itemsets.

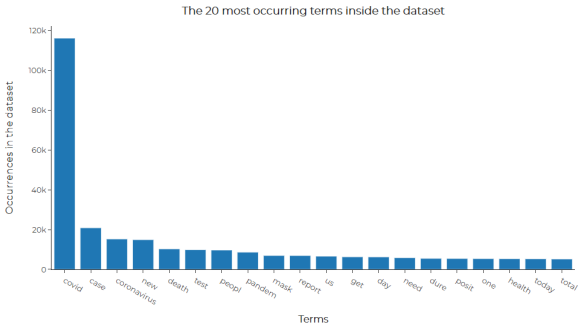

Figure 2: The 20 most occurring terms in the COVID19 dataset.

5 SOLUTION

6 IMPLEMENTATION

7 EXPERIMENTAL EVALUATION

REFERENCES