SocialPulse: a Tool for Extracting Interesting Insights from Social Media

Earass Ahmad and Kifayat Ullah Khan

November 8, 2022

SocialPulse: A Tool for Extracting Interesting Insights from Social Media

Earass Ahmad
Department of Data Science
FAST NUCES
Islamabad, Pakistan
i191254@nu.edu.pk

Kifayat Ullah Khan
Department of Data Science
FAST NUCES
Islamabad, Pakistan
kifayat.alizai@nu.edu.pk

***Abstract*—Digital media provides a huge amount of data. This data has rich content and gives us an opportunity to find interesting insights from it. The data consists of texts and other related attributes. The textual data can be considered as documents and hence extract valuable information from it. To find interesting insights, many approaches have been proposed so far. A shortcoming of such approaches is that the structure of the documents is neglected as the primary attribute remains the frequency. This, as a result, loses some of the valuable characteristics of the documents. In this work, we build a framework called Social Pulse that uses keywords to extract live tweets from Twitter and extracts multifold meaningful information from it. It is a complete framework that consists of a data pipeline that fetches and processes tweets, incorporates graph mining, has micro-services to serve data from backend to front-end, and provides a dashboard to visualize the analysis in the form of charts and graphs. At the core of the Social Pulse, we use gSpan, which is a famous and one of the most efficient Frequent Subgraph Mining (FSM) algorithms. We implement a parallel execution of gSpan in which we leverage the multicore processing technique to run gSpan in parallel to improve the execution time. The parallel implementation is imperative because the social media data grows large in size so the sequential run would take a lot of time to process. Our approach uses cooccurrence graphs to represent textual data in graphical form. The tweets' texts from Twitter are preprocessed and converted into co-occurrence graphs. The gSpan then extracts the frequent subgraphs from the graph database to infer the most common phrases occurring in the texts. Along with the tweet's text, there are multiple other attributes associated with the tweet. We use those attributes to infer multiple meaningful insights from the data.**

***Index Terms*—Interesting Insights, Social Pulse, Frequent Patterns, Text Analysis**

I. Introduction

A. Background

In this age of growing digital media, it takes almost no time for a topic to become a trend [4, 5], news to become breaking news [1], and discussions to become hot topics. The increased usage and easy accessibility of digital media across the globe helps people express their opinions about different aspects. This, as a result, intensifies the need of instant information gain through summarization of events. Topic modeling has since been used to extract the relevant hidden topics from the textual data. There are numerous related studies that have played their part in solving this problem. Some of them have used the NLP tech-niques that include LDA and LSI [17, 10, and 18]. Others have used the subgraph mining techniques [16] to explore this problem. The NLP based approaches have been explored extensively already. However, there is a potential to explore utilization of the subgraph mining for discovering interesting topics and phrases from the documents. Therefore, in our proposed approach, we use the frequent subgraph mining-based implementation to extract interesting phrases from text documents. Our proposed implementation represents texts as co-occurrence graphs and then passes the nodes and edges relationships to our proposed distributed gSpan algorithm for frequent topics and phrases extraction. The details of each step are provided in the later section. Also, apart from extracting most common phrases, we build other analytics using the twitter data. These analytics have a wide range and incorporates time, location, sentiments, users etc.

B. Motivation

Social media data analysis is imperative in having a glance at the current trends. The analysis of the data could help businesses, organizations as well as individuals. The applications range from user-timeline analysis to analyzing trends on digital media periodically. In order to build meaningful analysis, an effective framework is required. The framework should be an end-to-end system that is capable of pulling the data from the social media, processing it to derive insights and then visualize the insights in a way that is apprehended by a common person. In our work, we build such a scalable framework that is mentioned above. The primary objective of which is to derive meaningful insights from the raw social media data. These insights could be used for multifold purposes like business growth, current affairs knowledge, user profiling, and so on. For generating information from textual data alone, a significant amount of work has already been done in this primarily making use of the NLP techniques. However, we discovered that a huge potential lies in Subgraph mining approaches to extract information from the data. Subgraph mining is a vast field with so many state-of-the-art algorithms. Therefore, the efficiency of algorithms like gSpan for Frequent Subgraph Mining could be leveraged to extract the frequent terms and phrases from the documents. As we have discussed that there exist many efficient algorithms that use the NLP approaches but they have certain limitations. In order to address these limitations, we explored the domain of subgraph mining for the frequent topics extraction and generating other analytics. These algorithms offer such scalability that they could be tweaked to solve independent problems. The gSpan, for instance, makes use of the parallel processing for improved processing speed.

C. Problem Statement

The social media data provides a significant opportunity to find meaningful insights from it. Therefore, a complete end-to-end framework is required to generate and visualize these insights from the data. The core of this framework should be based upon some efficient algorithm, supported by data transformations that could extract the desired insights and then effectively present it to the user in the form of visualizations.

II. Literature Review

A. Graph Representations of Text Data

There are numerous ways in which textual data could be represented as graphs [13]. These include co-occurrence graphs, co-occurrence with POS tags, semantic graphs and others. Some of the representations suit some kinds of problems while others address some different kinds of problems. There have been multiple pa-pers published on ways to represent the textual data in the form of graphs so that the inherent characteristics of the text is preserved. Castillo et al. [8] in their paper, experiment with multiple co-occurrence graph-based representations for different text classification tasks. One of the graph-based representation is the star topology representation in Figure 1 which has a central vertex that is connected to every other vertex. The edges between vertices are formed if the words co-occur in a window size of two words. Also, the labels show the POS tags of the words. Another type of representation in the paper is for the sentiment analysis task Figure 1. It is a co-occurrence graph with a window size of three words. That means, two words if co-occurring in a window of three consecutive words would have an edge between them. The weight of the edge shows the number of times the two vertices co-occur.

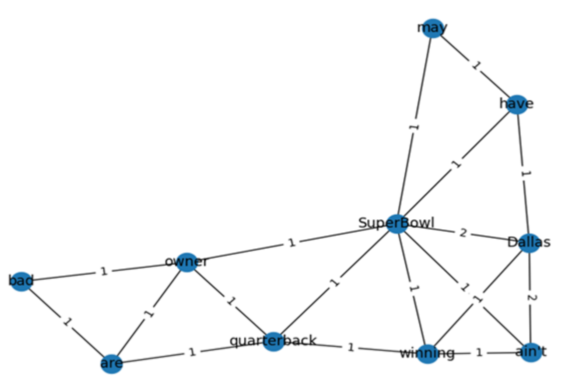

Fig. 1. Co-occurrence graph with window size of three words

Jiang et al [9] in their paper, also propose a graph representation of text data. They formulate a graph using four different types of nodes that include structural nodes, POS nodes, Token nodes that represent words and semantic nodes that represent additional information. The above discussed representations are statistical, that is, using the co-occurrence relationships. Some work has also been done on exploring the graph representations using linguistic relationships such as semantic and syntactic rela-tions. One such work has been done by Yadav et al. [11]. They propose an ap-proach to construct a semantic graph for text mining. Their approach consists of multiple sequential steps that are to stem the words, extract noun words only and find semantic relationships between words using WordNet. Each of the extracted noun token is treated as a node and a relationship between these tokens (nodes) is established if they have a semantic relationship. Apart from this, RK Rao et al. [12] use the Conceptual Graphs (CGs) for summa-rizing patent documents. They leverage the NER and other NLP techniques to extract concepts from the patent text documents. The writing style in patent doc-uments is complex and involves a lot of technical and legal concepts. This is one of major reasons for them to opt for this method.

B. Subgraph Mining

Some research work has been done on using sub graph mining for topic extrac-tion, detection, text summarization and clustering. Most of the problems related to topic extraction have been solved using the Frequent Subgraph Mining (FSM). The reason for it is that frequency is a very important element when extracting a topic out of a text document. Therefore, when text is represented as graphs, it is imperative to in-corporate the frequency of occurrence of words to identify which topics are under discussion in the document. Among many existing FSM algo-rithms, some of the state-of-the-art algorithms include gSpan, Gaston, MoFA, FFSM, SUBDUE, CloseGraph and others. A quantitative review has been done by Wörlein et al. [14] on the popular fre-quent subgraph miners. They provide stats that show that Gaston has a better execution time on the test datasets. However, when tested on large datasets, gSpan shows efficiency in the consumption memory and the time taken for processing.

Nguyen et al. [16] in their work leverage the co-occurrence graphs and gSpan algorithms to find the hidden topics in document dataset.

Zhao et al. [17] propose another graph-based model for detecting topics from a document database. Their approach considers more semantic relationships be-tween entities in-stead of relying solely on statistical approaches. They leverage the spectral clustering algorithm along with LDA to identify major topics from the dataset

Another implementation by Bekoulis et al. [10] use the LDA and LSI ap-proaches to detect topic from the dataset. Instead of the term frequency based LDA and LSI they propose a term weight based LDA and LSI approaches. This skips the traditional bag of words method and incorporates the importance of nodes in the network. They use the co-occurrence graphs for representing text data in the form of graphs.

Pham et al. [18] have proposed an approach called GOW-LDA. They create a graph-of-words, extract frequent sub-graphs and use LDA to detect topics from text datasets. The graph created is a co-occurrence graph with a window size of 3 words to capture the trigrams.

C. Parallel FSM

Some of the work has been done in running Frequent Subgraph Mining using a parallel implementation. These algorithms use various methodologies for distrib-uted implementation such as MapReduce, parallel processing using multicores, Apache Spark and so on.

Nguyen et al. [22] in their work propose a frequent subgraph mining approach to discover topics from large text documents. They use gSpan for finding frequent subgraphs and run the algorithm in parallel using Apache Spark. One of the drawbacks of this implementations is that when executing in parallel, it generates candidates. That is not happening in the original gSpan algorithm because it is a Pattern Growth algorithm that do not generate candidates for extracting sub-graphs.

Sangle et al. [23] propose a distributed gSpan algorithm called gSpan-H. They use the MapReduce framework to implement this. The algorithm generates frequent subgraphs without candidate generation.

Lakshmi et al [24] propose a distributed implementation of gSpan for frequent subgraph mining. They use the multicore technology to run the algorithm in par-allel. One of the shortcomings of their implementation is that it works only for the tree-based structure graphs that have a parent-child relationships among nodes.

Going through the related work above, we see that there is a need to explore the Frequent Subgraph Mining (FSM) for generating insights. There have been nu-merous implementations using NLP techniques as compared to a very few im-plementations based on FSM algorithms.

D. Data Analysis Tools

Béres et al. [25] build an interactive dashboard based upon the COVID data from Twitter. In particular, they generate the sentiments of people who voice their opinion on COVID vaccines. The dashboard shows sentiments with respect to the location and information type.

Gaglio et al. [26] propose a framework for analysis of real-time Twitter data. They tweak the Soft Frequent Pattern Mining (SFPM) algorithm for improve-ments. They organize the stream of tweets in a dynamic window which varies based on the volume and time of the tweets. The primary analysis they are doing is topic detection

Casalino et al. [27] propose a framework that discovers topics and groups tweets accordingly from the Twitter data. They use the non-negative matrix factorization technique to extract topics from tweets that are human interpretable. The use the word cloud and clustering to visualize the data in the end.

Wang et al. [28] use the Twitter data to analyze the COVID 19 health-related be-liefs of users. They used four constructs to build a model to quantify health be-lief. The machine learning and NLP based models were used to judge if the tweets conformed with the health belief model.

Shu et al. [29] create a tool for detecting fake news from social media. They make a complete framework the collects the data from social media, predicts which information is fake and then effectively visualizes that information in a dash-board. For detecting fake news, they build a model using deep LSTMs.

Sharma et al. [30] build a framework that identifies misinformation on covid re-lated tweets. The tool provides sentiments and topic clustering around covid re-lated hashtags and also detects if the information is false.

E. Existing Tools and Visualizations

Béres et al. [25] build an interactive dashboard based upon the COVID data from Twitter

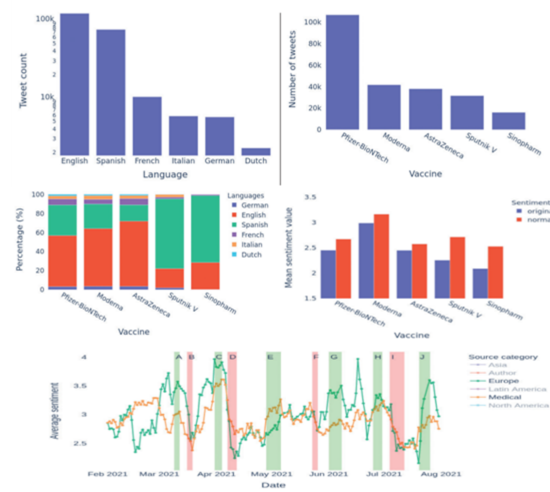

Fig. 2. Béres et al. [25] - COVID Vaccine Sentiment Dashboard

Wang et al. [28] use the Twitter data to analyze the COVID 19 health-related beliefs of users

Soundararaj et al. [31] build a dashboard to determine the im-pact of COVID on real estate using Twitter data.

III. Proposed Solution

We build a complete end to end framework called the Social Pulse which takes a keyword as an input from the user, fetches tweets against that keyword, gener-ates insightful data and then visualize it on the front-end dashboard. The details of this tool are mentioned in the later section. To address the problem of frequent words and phrases extraction from the tex-tual datasets, the solution proposed incorporates the co-occurrence graph based frequent subgraph mining approach. We propose the distributed gSpan algorithm. Distributed gSpan is an extension in gSpan in which we leverage the distributed computing to run gSpan in parallel to improve the execution time.

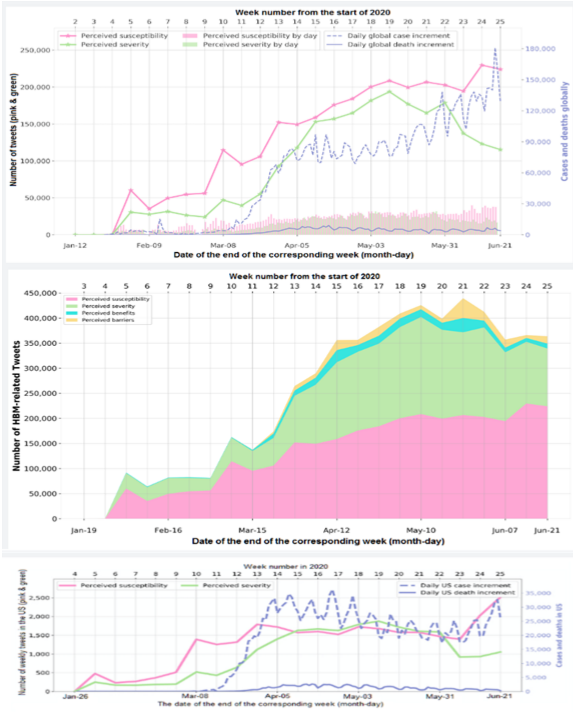

Fig. 3. Wang et al. [28] - Using Tweets to Understand How COVID-19–Related Health Beliefs Are Affected in the Age of social media: Twitter Data Analysis Study

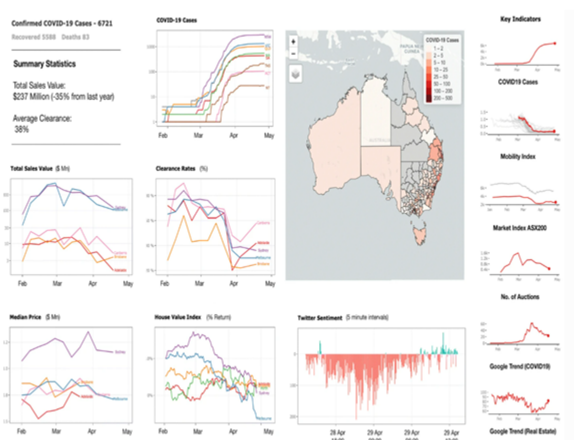

Fig. 4. Soundararaj et al. [31] - Property Dashboard

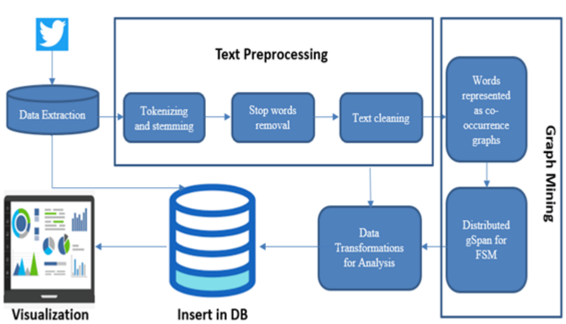

Fig. 5. Methodology

A. Data Extraction

The data extraction part includes fetching textual data from Twitter and YouTube videos transcripts. However, our model is not dependent on the data in the devel-opment phase as it does not require any training. Also, the model can run on any text documents and is not limited to Twitter or YouTube data alone.

B. Data Preprocessing

Textual data needs some preprocessing before it is passed on to any model. Some of the steps that we perform are:

- Stop words removal
- Word tokenization and stemming
- Remove punctuations if causing noise

C. Graph Representation of Text

The graphical representation of textual data we use is the co-occurrence graphs [19]. In this graph, each word in the document would be a vertex of the graph. There would be an edge between two nodes (words) if they appear together within a specified window size in the document. The weight of the edges shows the number of times the two words have appeared together. In our approach, we use a non-directed co-occurrence graph with a window size of three words. So, the proposed graph G is represented by

$$G = (V, E, L_V, L_E) \tag{1}$$

Where V is the finite set of vertices, E is the finite set of edges which represent the vertices are connected if they appear together in a window size, LV is the label set of V and LE is the label set of E, the number of times two vertices have co-occurred.

D. Frequent Subgraph Mininig

Frequent Subgraph Mininig (FSM) is a technique in graph mining to extract frequent occuring subgraphs from the graph dataset. In our situation, the textual data is represented in the form of graphs so that the frequent occurring subgraphs, which in this case would be frequent phrases, could be extracted. When the text data is in the form of co-occurrence graph, each subgraph would be a phrase of text from the original document. Therefore, the extracted frequent subgraphs would represent the most common occuring phrases of text from the documents.

E. gSpan

gSpan, short for graph-based Substructure pattern mining was introduced by Yan et al. [21] to extract frequent subgraphs from graph databases. It is one of the most efficient algorithms for finding frequent subgraphs. gSpan is used widely because of its efficient use of resources to find frequent subgraphs. It reduces the search space by efficiently removing the unsuitable subgraphs from the branches of the DFS Code Tree. gSpan is a pattern growth-based approach and generates a structure that is like a tree. The tree is basically a Depth First Search (DFS) code tree for all the subgraphs. In that tree, every vertex represents a DFS code. The algorithm uses the Depth First Search strategy and the minimum DFS code based on lexicographic ordering to generate frequent subgraphs.

F. Distributed gSpan

gSpan is widely used frequent subgraph mining algorithm and has a very good efficiency. However, one of the limitations of gSpan is the execution time. The algorithm takes a lot of time if run on the large graph datasets. Therefore, we make use of the parallel processing to extract frequent subgraphs from large graph databases using gSpan. We parallelize the process of growth from single edge to mine subgraphs.

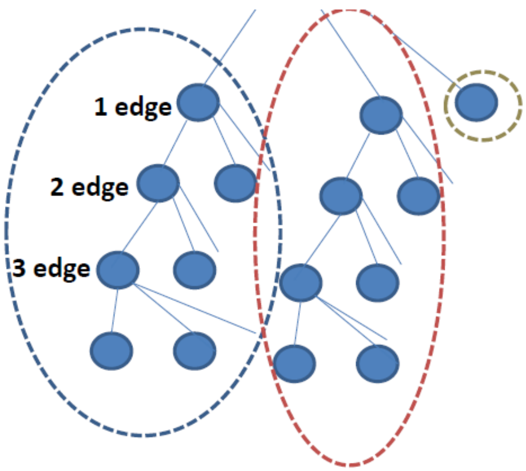

Fig. 6. Parallel FSM

G. Data Transformations

At this stage, we transform our data for the purpose of generating interesting insights using the information we have. We perform data transformations to extract the following:

- Classify each text document as Hateful or not. For this purpose, we build a classifier that classifies if the text contains hate speech or not. The classifier was build using Deep Neural Network using approximately 50000 labelled tweets for training. It achieved the accuracy of 81 percent on testing data.
- Classify each text to be Positive, Negative or Neutral. For this sentiment analysis, we use an existing Python's library called TextBlob
- Most common words
- Location wise sentiments
- Hourly sentiments
- Hourly volume
- Hourly top users' volume
- Users with highest retweets
- Top mentioned users
- Top mentioned hashtags

H. Data Loading

After generating the data with insights, we insert it into MySQL tables so that this data could be used later.

I. Visualization

For visualizing the insights from the data, we create a web-based dashboard that contains multiple charts. User gives a keyword as input from the front-end. The keyword is passed to the backend service that fetches tweets from Twitter that contain this keyword. The tweets then pass through the above explained pipeline and generates the data which is then visualized on the front-end as charts. The fig below shows the layout of the dashboard.

IV. Experiments and Results

Following are some of the interesting insights that we generate as a result of our above pipeline. The below charts show the analysis from a sample of tweets on keyword 'Pakistan' obtained on 5/21/22 at around 21 hours.

The below figure shows the frequent subgraphs generated from the distributed gSpan. These subgraphs indicate that these phrases were common in multiple text documents which in this case are tweets. The frequent subgraphs are extracted using the minimum support of 2. That means that a subgraph is said to be fre-quent only if it appears in at least 2 graphs. The node labels in the subgraphs be-low show that these phrases were commonly present in multiple tweets.

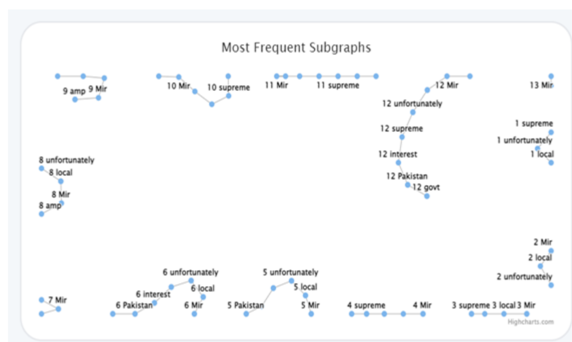

Fig. 7. Frequent Subgraphs

The below word cloud shows the most frequent words that appeared in the text documents. These words were extracted from the tweets text, cleaned, lemma-tized and then the most common were extracted.

Fig. 8. Frequent Words

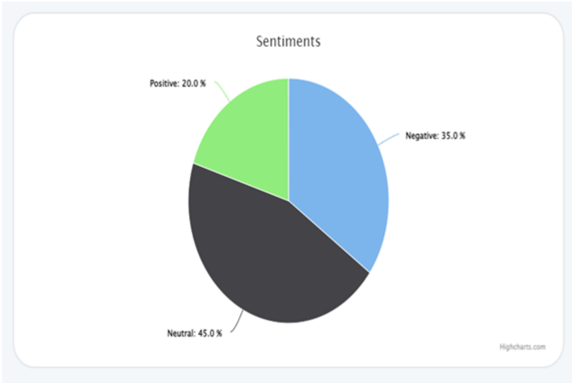

Fig. 9. Sentiments

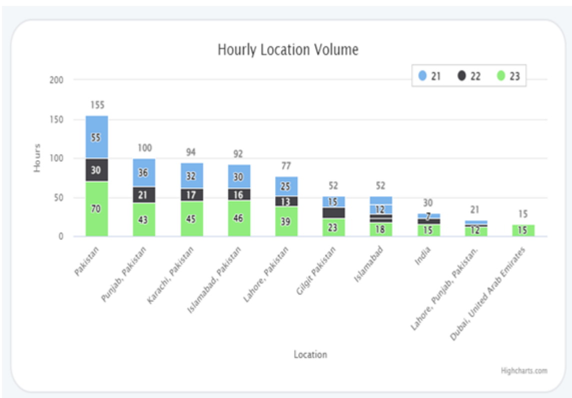

Fig. 10. Hourly Location Volume

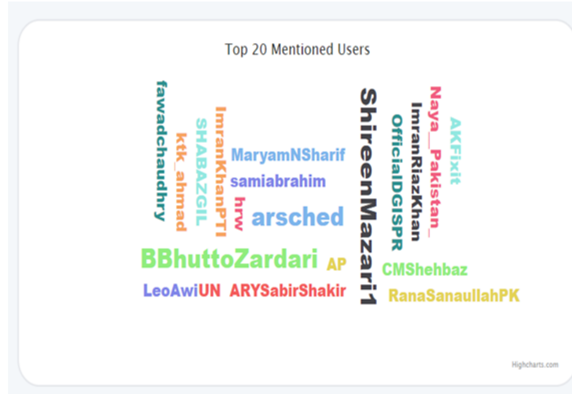

Fig. 12. Most Frequent mentioned users

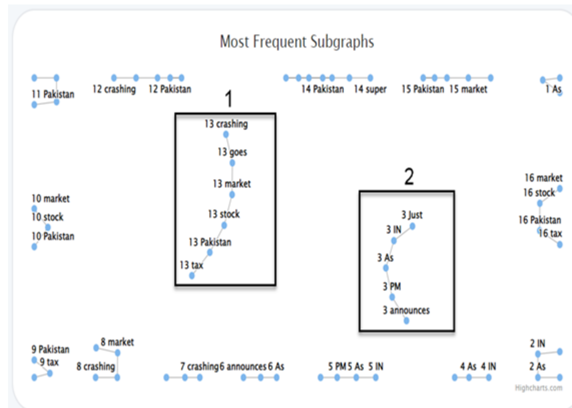

Fig. 13. Frequent Extracted Subgraphs

V. Evaluation

Our algorithm extracts the most common words and phrases from the text docu-ments. The evaluation is done by human interpretability to see if the results i.e., the words and phrases actually make sense or not.

Following fig shows some of the frequent subgraphs gener-ated as the output from our algorithm.

For the highlighted subgraph 1, the phrase becomes some-thing like “Pakistan stock market goes crashing tax”. This could be interpreted as news about the stock market crashing due to some tax laws.

For the highlighted subgraph 2, the phrase becomes some-thing like “Just in as PM announces”. This could also be interpreted as some announcement from PM that has caused the stock market to crash.

Looking at the above, we see that the phrases do make sense and are under-standable by humans.

For the evaluation of distributed execution of gSpan, we evaluated two things:

- The extracted frequent subgraphs were the same as that from the sequen-tial run.
- During the experiments we have run the gSpan with parallel and series implementations. The number of doc-uments were changed and the other parameters were kept constant. The minimum support was set to be 2. Following were the observations in this experiment.

For the smaller document sizes, we see that the time for series implementation is lower. This is because for parallel processing we come across an overhead that can be overcome when we have suitable processing or the size of the database is large enough so that the parallel processing actual benefits us.

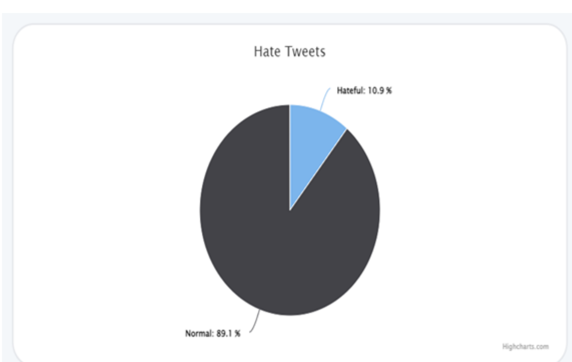

Fig. 11. Hate Speech Classification

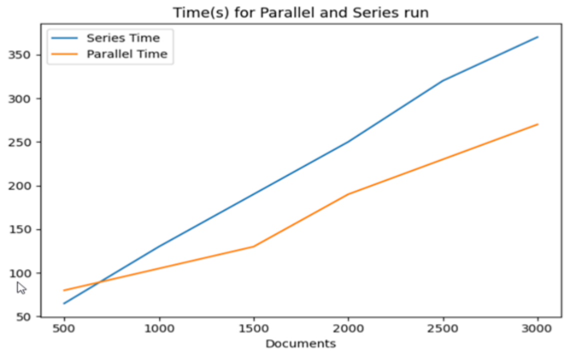

Fig. 14. Time taken by series and parallel runs for multiple document sizes

VI. Conclusion and Future Work

In this work, we created a complete end-to-end data analysis framework called Social Pulse. The framework is capable of extracting Twitter data, processing it to generate insights and then visualizing those insights on the front-end dashboard. Our core analysis is based on extracting frequent words and phrases from the tweets text for which we use the Frequent Subgraph Mining. This is supported with multiple other insights which carry a significant information. The applica-tion is structured in a way that it can easily be scaled with time.

In future, we would want to integrate multiple other social media platforms with our tool. These include Facebook, YouTube, LinkedIn and so on. This would achieve a horizontal scalability leading to a diverse analysis from the social media.

References

[1] Sarah A. Alkhodaira , Steven H.H. Dingb , Benjamin C.M. Fung∗,b , Jun-qiang Liuc, "Detecting breaking news rumors of emerging topics in social media", Information Processing and Management Volume 57, Issue 2, March 2020, 102018

[2] Onur Varol , Emilio Ferrara, Filippo Menczer, and Alessandro Flammini, "Early detection of promoted campaigns on social media", EPJ Data Sci-ence (2017) 6:13 DOI 10.1140/epjds/s13688-017-0111-y

[3] Zhifan Yang, Chao Wang, Fan Zhang, Ying Zhang, Haiwei Zhang, "Emerging Rumor Identification for Social Media with Hot Topic Detection", 2015 12th Web Information System and Application Conference (WISA)

[4] Rashmi Pokharel and Pari Delir Haghighi, "Analysing Emerging Topics across Multiple Social Media Platforms", ACSW 2019: Proceedings of the Australasian Computer Science Week Multi conference January 2019 Article No.: 16Pages 1–9https://doi.org/10.1145/3290688.3290720

[5] Hwi-Gang Kim, Seongjoo Lee, Sunghyon Kyeong, "Discovering hot top-ics using Twitter streaming data social topic detection and geographic clustering", 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)

[6] ZHILIANG ZHU, JIE LIANG, DEYANG LI , HAI YU , AND GUOQI LIU, "Hot Topic Detection Based on a Refined TF-IDF Algorithm", IEEE Ac-cess (Volume: 7) 2019

[7] Yan Chen, Hadi Amiri, Zhoujun Li, Tat-Seng Chua, "Emerging Topic De-tection for Organizations from Microblogs", SIGIR '13: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval July 2013 Pages 43–52

[8] Castillo, Esteban, Ofelia Cervantes, and Darnes Vilarino. "Text Analysis Using Different Graph-Based Representations." Computación y Sistemas 21.4 (2017): 581-599.

[9] Jiang, C., Coenen, F., Sanderson, R., and Zito, M. (2010). Text classification using graph mining-based feature extraction. In Research and Development in Intelligent Systems XXVI (pp. 21-34). Springer, London.

[10] Bekoulis, Giannis, and François Rousseau. "Graph-based term weighting scheme for topic modeling." 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). IEEE, 2016.

[11] Yadav, Chandra Shekhar, Aditi Sharan, and Manju Lata Joshi. "Semantic graph based approach for text mining." 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT). IEEE, 2014.

[12] RK Rao, Pattabhi, and Sobha Lalitha Devi. "Patent document summariza-tion using conceptual graphs." International Journal on Natural Language Computing (IJNLC) Vol 6 (2017).

[13] Sonawane, Sheetal S., and Parag A. Kulkarni. "Graph based representation and analysis of text document: A survey of techniques." International Journal of Computer Applications 96.19 (2014).

[14] Wörlein, Marc, et al. "A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston." European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, 2005.

[15] Güvenoglu, Büsra, and Belgin Ergenç Bostanoglu. "A qualitative survey on frequent subgraph mining." Open Computer Science 8.1 (2018): 194-209.

[16] Nguyen, Tri, and Phuc Do. "Topic discovery using frequent subgraph mining approach." International Conference on Computational Science and Technology. Springer, Singapore, 2017.

[17] Zhao, Ang, Xin Lin, and Jing Yang. "Graph-based Model for Topic De-tection." Proceedings on the International Conference on Artificial Intel-ligence (ICAI). The Steering Committee of The World Congress in Com-puter Science, Computer Engineering and Applied Computing (World-Comp), 2014.

[18] Pham, Phu, Phuc Do, and Chien DC Ta. "GOW-LDA: Applying Term Co-occurrence Graph Representation in LDA Topic Models Improvement." International Conference on Computational Science and Technology. Springer, Singapore, 2017.

[19] Hassan, S., Mihalcea, R., Banea, C.: Random walk term weighting for improved text Classification. Int. J. Semant. Comput. 1(4), 421–439 (2007)

[20] Serrano, M. Ángeles, Marián Boguná, and Alessandro Vespignani. "Extracting the multiscale backbone of complex weighted networks." Proceedings of the national academy of sciences 106.16 (2009): 6483-6488.

[21] Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: Proceedings of the IEEE International Conference on Data Mining, 2002, ICDM 2003, pp. 721–724. IEEE (2002)

[22] Nguyen, Tri, and Phuc Do. "Topic discovery using frequent subgraph mining approach." International Conference on Computational Science and Technology. Springer, Singapore, 2017.

[23] Sangle, Mrs MH, and P. S. A. Bhavsar. "gspan-h: An iterative mapreduce based frequent subgraph mining algorithm." International Journal of Advance Research and Innovative Ideas in Education 2.5 (2016): 169-177.

[24] Lakshmi, K., and T. Meyyappan. "Efficient algorithm for mining frequent subgraphs (static and dynamic) based on gSpan." International Journal of Computer Applications 63.19 (2013).

[25] Béres, Ferenc, et al. "COVID Vaccine Sentiment Dashboard based on Twitter Data." Scientia et Securitas 2.4 (2022): 418-427.

[26] Gaglio, Salvatore, Giuseppe Lo Re, and Marco Morana. "A framework for real-time Twitter data analysis." Computer Communications 73 (2016): 236-242.

[27] Casalino, Gabriella, et al. "A framework for intelligent Twitter data anal-ysis with non-negative matrix factorization." International Journal of Web Information Systems (2018).

[28] Wang, Hanyin, et al. "Using tweets to understand how COVID-19–Related health beliefs are affected in the age of social media: Twitter da-ta analysis study." Journal of medical Internet research 23.2 (2021): e26302.

[29] Shu, Kai, Deepak Mahudeswaran, and Huan Liu. "FakeNewsTracker: a tool for fake news collection, detection, and visualiza-tion." Computational and Mathematical Organization Theory 25.1 (2019): 60-71.

[30] Sharma, Karishma, et al. "Coronavirus on social media: Ana-lyzing misin-formation in Twitter conversations." arXiv preprint arXiv:2003.12309 (2020).

[31] Soundararaj, Balamurugan, Christopher Pettit, and Oliver Lock. "Using Real-Time Dashboards to Monitor the Impact of Disruptive Events on Real Estate Market. Case of COVID-19 Pandemic in Australia." Computational Urban Science 2.1 (2022): 1-13.