SANCTUM

Scalable Analysis on a Cluster of Twitter-Data Using Mining Strategies

Project Proposal

Shang-Ping Dai
Computer Science Honours
University of Cape Town
Cape Town, South Africa
dxxsha001@myuct.ac.za

Pieter van der Walt
Computer Science Honours
University of Cape Town
Cape Town, South Africa
vwlpie004@myuct.ac.za

Matt Young
Computer Science Honours
University of Cape Town
Cape Town, South Africa
yngmat005@myuct.ac.za

Keywords

Data mining, social media, cluster environment, Twitter, scalability, parallel processing, information retrieval

1 PROJECT DESCRIPTION

Social media produces vast quantities of data on a daily basis [1]. Both researchers and businesses can gain valuable insight into the dynamics of online communities by mining this data. Data mining is the process of generating insightful knowledge by examining very large sets of data [2]. The application of data mining to social media can yield benefits like gauging user sentiment for proactive planning and developing effective recommendation systems. It could also be used to identify influential individuals in communities [1]. However, these data sets are often very complex, because of their unstructured nature [4]. Additionally, the sheer size of the data sets meant that traditional, single-machine systems could not perform the required mining [4]. New systems that utilize clusters of computers to perform the tasks in parallel were developed [2]. The parallel nature of these systems required traditional data mining and information retrieval (IR) algorithms to evolve [3, 4]. The goal of this project is to build a system that can utilize a cluster to do a scalable analysis of large amounts of Twitter data. More details regarding the data set and the cluster environment is discussed in Section 3. The system will be divided into three logical sections, which will combine to form the whole system. The sections are:

1. A parallel information retrieval algorithm for the Twitter data.
2. An association rule mining algorithm that can process the Twitter data in parallel.
3. A Web-based user interface that will let users run queries and manage data mining jobs on the cluster. The Web interface will also feature scalable visualizations of the data mining results.

2 PROBLEM STATEMENT

The main problem we're attempting to solve is the need to extract useful information and generate insightful knowledge from a huge set of Twitter data in a timely manner. To do this we'll develop a system that performs efficient parallel information retrieval and association rule mining on a Twitter data set in a cluster environment. Information retrieval refers to the process of extracting and collecting specific information from stored data. Association rule mining is the procedure of detecting correlations, frequent patterns, and associations in a data set. The main issues that need to be resolved for the project to be a success are the following:

1. Figuring out how and where the data can be staged.
2. Identifying the necessary pre-processing procedures, and successfully implementing them.
3. Implementing an efficient parallel information retrieval algorithm.
4. Implementing a scalable parallel association rule mining algorithm.
5. Creating useful visualizations to represent the data mining results in a meaningful manner.
6. Building a user interface that researchers can use to interact with the system.

Our users will be any researchers who wish to utilize the cluster to mine the available Twitter data. We have consulted some potential users and have gathered preliminary requirements. These requirements are not final and will be scrutinized and refined later.

Back-end	
1	Provide pipelining capabilities for the staging, pre-processing and analysis of data.
2	Expectations of queries and data mining jobs to complete in an order of minutes. Very dependent on job size.
Front-end	
1	The homepage of the interface should display the following information: the

	specifications and current configuration of the hardware, possibly an estimated time to completion for jobs that are still in progress, and a log of successfully completed jobs.
2	The interface should provide a method for ring-fencing the data through the execution of queries. These queries include searching for hashtags, usernames, phrases, or individual words.
3	Display data mining results in list format with computed summaries. Also display appropriate graphs for the job-type. For example, a line-graph displaying the change in user sentiment towards a subject over a period of time.
4	Be able to export the data mining results in XML format.

Table 1. Preliminary gathered requirements

3 PROCEDURES AND METHODS

We look into building a system that can efficiently find patterns in this data. The system consists of various pipelined processes that can be applied in parallel on a cloud computing platform to address the scalability issues that one will face when dealing with this quantity of data.

3.1 Dataset

3.1.1 Twitter Data Format

The Twitter data to be processed was obtained by our supervisor, Prof. Hussein Suleman, and is 30 terabytes large.

All Twitter APIs that extract Twitter data require the data to be encoded in JavaScript Object Notation (JSON) [11]. This format is based on key-value pairs with named attributes and associated values. Each tweet consists of an author, a message, a unique ID, a timestamp of when it was posted and sometimes geo metadata shared by the user. Each tweet object also generates an `entity` object, that is an array of common tweet contents such as hashtags, mentions, media and links.

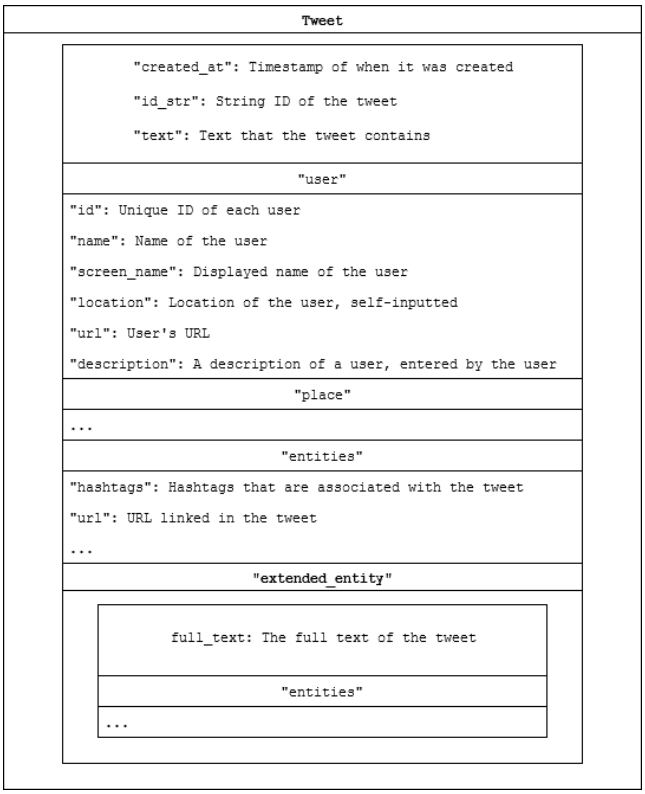


Diagram 1. General format of a tweet

Tweet data is structured as shown in diagram 1 above. Each tweet consists of basic fundamentals as discussed, but now also include an `extended_entities` object. This object was newly implemented by Twitter in a recent update that supported a new 280-character limit. This object is used to essentially contain a tweet inside a tweet and was implemented in this way was to encapsulate these longer messages, while not breaking the thousands of apps parsing these fundamental Twitter objects [11].

3.1.2 Sample Datasets

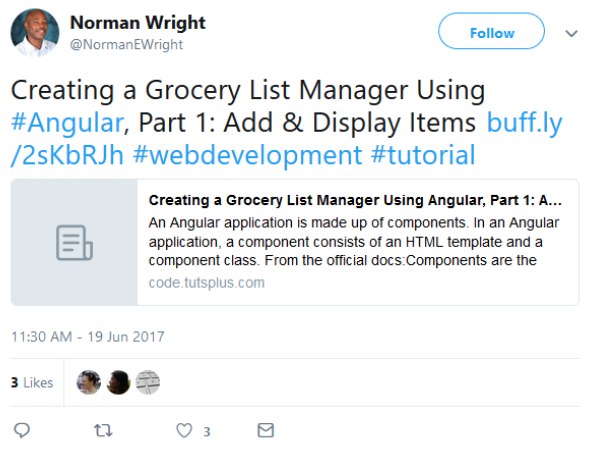
Below are a few examples of tweets and their corresponding JSON representations. The JSON representations will only contain some of the objects of the tweet.

Example tweet 1.


```
{
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
  "id_str": "850006245121695744",
  "text": "1\/ Today we\u2019re sharing our vision for the future of the Twitter API platform!\nhttps:\/\/t.co\/XweGngmxlP",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https:\/\/dev.twitter.com\/",
    "description": "Your official source for Twitter Platform news, updates & events. Need technical help? Visit https:\/\/twittercommunity.com\/ \u2328\ufe0f #TapIntoTwitter"
  },
  "place": {
  },
  "entities": {
    "hashtags": [
    ],
    "urls": [
      {
        "url": "https:\/\/t.co\/XweGngmxlP",
        "unwound": {
          "url": "https:\/\/cards.twitter.com\/cards\/18ce53wgo4h\/3xo1c",
          "title": "Building the Future of the Twitter API Platform"
        }
      }
    ],
    "user_mentions": [
    ]
  }
}
```


JSON document representing example tweet 1.

Example tweet 1 shows a tweet that has linked a URL with text.



Example tweet 2.


```
{
  "created_at": "Thu Jun 22 21:00:00 +0000 2017",
  "id": 877994604561387500,
  "id_str": "877994604561387520",
  "text": "Creating a Grocery List Manager Using Angular, Part 1: Add &amp; Display Items https://t.co/xFox78juL1 #Angular",
  "truncated": false,
  "entities": {
    "hashtags": [{
      "text": "Angular",
      "indices": [103, 111]
    }],
    "symbols": [],
    "user_mentions": [],
    "urls": [{
      "url": "https://t.co/xFox78juL1",
      "expanded_url": "http://buff.ly/2sr60pf",
      "display_url": "buff.ly/2sr60pf",
      "indices": [79, 102]
    }]
```

```
 },
 "source": "<a href=\"http://bufferapp.com\"
rel=\"nofollow\">Buffer</a>",
  "user": {
    "id": 772682964,
    "id_str": "772682964",
    "name": "SitePoint JavaScript",
    "screen_name": "SitePointJS",
    "location": "Melbourne, Australia",
    "description": "Keep up with JavaScript
tutorials, tips, tricks and articles at
SitePoint.",
    "url": "http://t.co/cCH13gqeUK",
    "entities": {
      "url": {
        "urls": [{
          "url": "http://t.co/cCH13gqeUK",
          "expanded_url":
"http://sitepoint.com/javascript",
          "display_url":
"sitepoint.com/javascript",
          "indices": [0, 22]
        }]
      },
      "description": {
        "urls": []
      }
    },
    "protected": false,
    "followers_count": 2145,
    "friends_count": 18,
    "listed_count": 328,
    "created_at": "Wed Aug 22 02:06:33 +0000
2012",
    "favourites_count": 57,
    "utc_offset": 43200,
    "time_zone": "Wellington",
  },
}
```


JSON document representing example tweet 2.

Example tweet 3 shows a more in-depth look at the children objects of the link. The tweet also includes hashtags and details of the user who posted the tweet.

3.2 Hadoop

We will make use of Apache's Hadoop library[1] to implement the parallelized IR of Twitter data and a distributed implementation of the Apriori algorithm for association rule mining over MapReduce. Hadoop provides a distributed programming framework that allows for the processing of large amounts of data and provides a framework for the MapReduce model in which Apriori can be implemented.

MapReduce clusters have one master node and many worker nodes [12]. When a job is initialized, the master node will schedule map functions and reduce functions to various nodes. An advantage of the MapReduce framework is providing sufficient scalability. This is done by further partitioning the data set and simply scheduling more map or reduce functions as the data set grows.

3.3 Information Retrieval

Information retrieval is the finding of material of an unstructured nature from large collections that satisfies an informational need [3].

3.3.1 System Description

The Information Retrieval subsystem is responsible for fetching and indexing data segments from a large data source based on a user query.

The system will be a multithreaded architecture that performs two major tasks:

- Querying - involves processing the user request from the front-end / user interface to fetch the desired data.
- Indexing - involves organizing the queried data in a data structure, which can be accessed by the pattern recognition system.

The system will utilize a distributed information retrieval algorithm that can efficiently retrieve data from UCT's High Performance Cluster. In order to do so, it will make use of the Apache Hadoop framework (discussed earlier).

3.3.2 Evaluation

In order to test the system, we have chosen the following performance measures:

- Computation time: the overall running time to complete an information retrieval task
- Speedup: the factor at which the computation time increases as the data input size increases

[1] The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. Available at: http://hadoop.apache.org/

- Correctness comparison: using an existing testbed of established reliable data, a comparison between this system's output and the existing output will show a degree of correctness in our system.

3.4 Association Rule Mining

Following the retrieval of relevant tweets based on some topic or hashtag, a proposed method of finding patterns is by using association rule mining. Although the retrieval of more relevant tweets fetches a subset of the entire dataset, the retrieved data set would still be too large for sequential rule mining to fully generate rules from. In past research [15], association rule mining can be done more viably by using a cluster. The standard Apriori algorithm [9] can help develop rules between a basket-like representation of tweets and has been implemented in parallel on a cluster before in past research [15].

Firstly, the twitter data must be preprocessed into a basket-like representation, so that each tweet will be represented as a list of topics that the tweet is focused on. Once tweets are represented as a list of topics, we will then be able to generate a set of rules using Apriori.

These rules are found by first defining a support and confidence threshold. The support threshold tells the algorithm to only develop rules for those topics or words that are popular enough. This ensures that rules are only generated for topics that are tweeted often. The confidence signifies the likelihood of topics in a rule being in the same tweet. This is used when generating rules as only rules that exceed the confidence threshold will be outputted as results.

The rules that Apriori generate relate a set of topics in a tweet to another set of topics in a tweet, and this relation will imply that if a tweet contains a topic or word from the first set, it is highly likely that it will contain a topic from the second set.

3.4.1 Algorithm Implementation

The preprocessing of tweets can be done by using a topic detection algorithm to extract topics from tweets such that each tweet can be represented as a list of topics that the tweet is focused on.

Once this has been done, a scalable version of Apriori will be called to generate rules on the datasets. The standard Apriori algorithm consists of two steps. The first step is the generation of frequent itemsets. Frequent itemsets are sets of items, or topics in this context, that are considered the most popular among the tweets. These are found by counting the occurrences of each item in the dataset and determining whether the percentage of occurrence exceeds that of the predefined support threshold. The second step of Apriori is the generation of association rules. A scalable version of Apriori will be developed by implementing slower performing steps of the Apriori algorithm onto Hadoop's MapReduce framework discussed earlier. This step would be the generation of frequent itemsets. The map and reduce functions will be written to implement the generation of frequent itemsets, and rules will then be generated from these itemsets.

3.4.2 Evaluation

The aim of the generation of association rules is to present these frequent patterns in usable times to researchers so the evaluation of the rule mining section will be similar to the IR section: with respect to scalability and computation time. This is done by evaluating the following:

- Execution times of the program as the dataset increases
- Speedup of the algorithm as the dataset increases
- Correctness of the algorithm using existing testbeds of established, reliable data

3.5 User Interface and Visualization

3.5.1 Interface Implementation

The Web interface will be developed in Javascript. This is because the language is accepted and supported by all major browsers. It allows both client-side and server-side scripts to be executed, which is crucial to for the system to function. Javascript also has a large number of libraries available for use. The ExpressJS[2] Web framework will be used to build the interface. The framework will utilize the Hadoop YARN Web service REST APIs[3] to interact with the Hadoop cluster. These APIs, along with other secure communication protocols will be used to execute custom scripts on the cluster. The website will be hosted on a UCT server. The design will be based on the requirements outlined by the potential users in Section 2. The interface will be repeatedly tested with potential users throughout the development process. This will ensure that the interface and visualizations are optimized for usability and usefulness. The interface will require users to login. The server that hosts the website will store a database that contains the list of users with access to the system. Once logged in, users will be directed to the system's home page, where they will be able to access the tools and functionality built into the system.

3.5.2 Queries and Visualizations

The interface will feature a tool that lets users select a data set, and input the combination of keywords, hashtags and/or usernames they'd like to search for. These search terms will be relayed to the cluster using the above-mentioned protocols. Query results will be displayed in a table format. Association rules will be visualized as a grouped matrix with the ability to zoom in on specific areas if there's a large number of rules to display.

---

[2] ExpressJS is a free and open-source Web application framework built for the Node.js platform

[3] These are a set of URI resources that give access to applications and nodes on the cluster, as well as historical information.

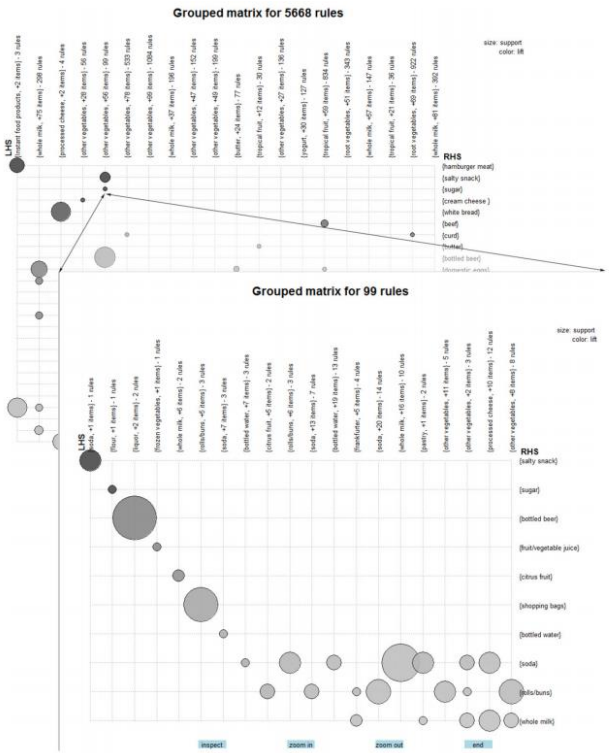


Figure 1. A grouped-matrix based visualization with the ability to zoom in on particular zones [14]

If there is enough time, the Information Retrieval system will also feature sentiment analysis capabilities, in which case visualizations such as the one below could be built as well.

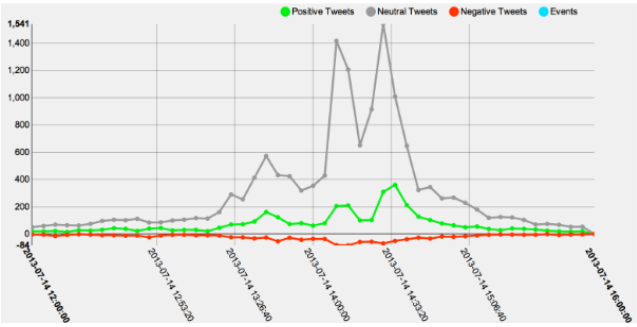


Figure 2. A timeline visualization of 4 hours of Twitter data for a user-specified query, aggregated on 5-minute intervals [10]

3.5.3 Evaluation

The interface will be evaluated based on its usability, usefulness, and user experience. Users will be asked to interact with the interface by performing a set of tasks. They will also be allowed to freely explore the platform. The evaluation will be based on Nielsen's heuristics [4]and other standard HCI measurements. Note that the evaluation will be repeated more than once during the development process, not only at the end.

3.6 Development Strategy

The design and development process will follow an iterative programming paradigm, more specifically Rapid Application Development (RAD). RAD supports meeting deadlines and reduces the risk of running off schedule. Incremental development will help reduce bugs and support test-driven development. It will also allow us to produce a workable prototype early. By having workable prototypes, we will be able to build a higher-quality final product.

3.7 System Diagram

The proposed system is represented as a system diagram shown in figure 3.

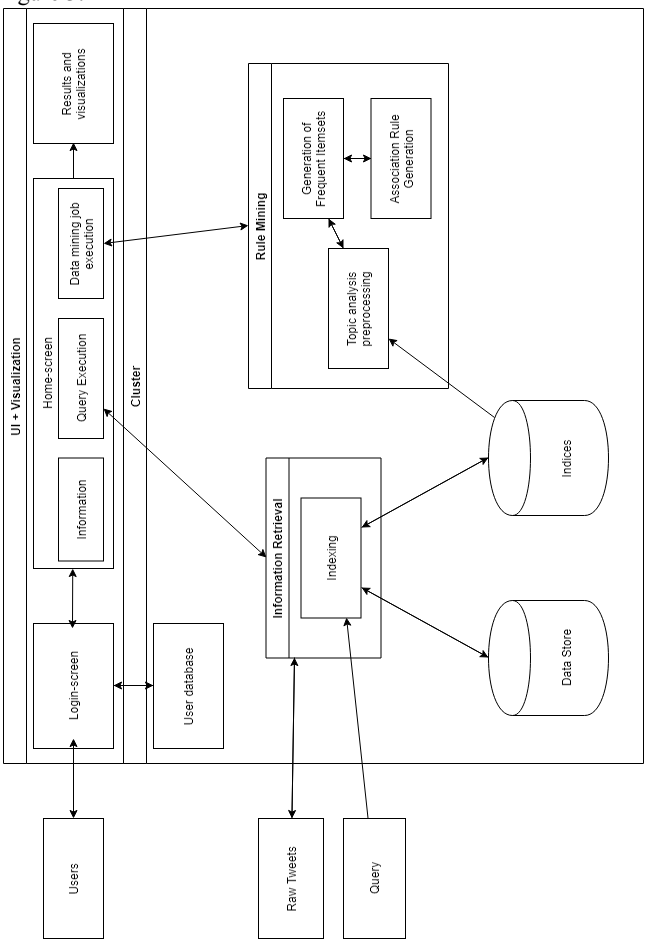


Figure 3. Proposed system diagram

[4] Jakob Nielsen is a Danish web usability consultant. He developed the 10 heuristics together with Rolf Molich in the early 90's.

4 ETHICAL, LEGAL AND PROFESSIONAL ISSUES

Due to the influx of social media data, topics on ethical and privacy issues have been widely discussed. This is due to the nature of social networks - people should have a sense of control as to how their data is accessed and used.

4.1 Privacy Protection

These days, a large portion of individuals' social, communicative, and private actions take place on digital media, and persists on these media [5]. This means that measures taken to protect user privacy have to be carefully defined. A data mining system like ours has to preserve the integrity of its source data to prevent any privacy issues from arising. In order to do so, there are two different privacy breaches that must be considered: social link disclosure and affiliation link [5]. Social link disclosure refers to the ability to identify sensitive or private relationships that were meant to stay hidden. Affiliation link disclosure refers to the ability to identify whether somebody is affiliated with a particular group.

4.2 Ethical Issues

Closely tied to privacy protection, ethical issues usually occur when private data is misused or exploited for personal gain. An example of this is the selling of personal data to large businesses for marketing purposes. Unless stated in a terms of use agreement, this is considered a breach in the user's trust. Our system is not being built to gain some sort of competitive business advantage for its users.

5 RELATED WORK

Social media platforms offer a data-rich environment, where data can be processed to find various unique or unexpected patterns, usually with data mining techniques. A major challenge that researchers face when attempting to find these patterns is first finding an efficient way of processing such large amounts of data that these social media platforms, such as Twitter, usually have. An example of similar research includes a pattern-based topic analysis system based on Zhang et al [8]. Their developed system analyzed Sina Weibo data, a Chinese social media platform similar to Twitter. The system also made use of a pipelined approach where the social media data was first preprocessed before finding patterns using a pattern mining algorithm. This preprocessing was done so that unnecessary words could be filtered out to perform some sort of topic analysis and detection on the data. Their research, however, did not focus on the scalability of the system.

Farzanyar et al [7] investigated the scalability of processing large amounts of social media data. Their paper focused on dealing with the scalability and efficiency of finding patterns in large social media data. They implemented the generation of frequent itemsets step of the standard Apriori algorithm for association rule mining in the MapReduce framework. The proposed algorithm was used on a cluster platform using Apache's Hadoop framework. The algorithm was tested on almost 520,000 transactions and was evaluated on it's computation time. The research managed to achieve execution times ranging from a minimum of 200 seconds to a maximum of 1200 seconds by varying the support threshold.

Hoeber et al. [10] proposed a program, called Vista, that could visually represent changing sentiment in Twitter data about sporting events. The program identified the temporal nature of tweets as an important aspect and thus decided to use a timeline as the core visual representation. Tweets associated with three different classifications of sentiment (positive, neutral and negative tweets) were given unique colors and the number of tweets that had these sentiments were plotted on the timeline to represent the divergent nature of these sentiments. Vista was also able to drill down on specific parts of the timeline.

Hahsler et al [14] also explored various methods for visualizing patterns from association rules in a hierarchy. These attempts can be classified as either matrix-based visualization, graph-based visualization, or grouped-matrix based visualization.

6 ANTICIPATED OUTCOMES

6.1 System Outcomes

We propose to build a modularized system composed of the following subsystems:

- Information Retrieval Module
- Association Rule Mining Module
- Web Front-End Tool

The Information Retrieval Module is responsible for indexing and querying the Twitter data on a large scale. The Association Rule Mining Module holds the implementation of the parallel data mining/pattern recognition algorithms. The input of the Association Rule Mining Module is partially dependent on the indexed IR data to generate correct results; however it can still run on any data set. The Web Front-End Tool implements a user interface to interact with the other two subsystems. It is used to query the data set, generate associations and patterns, and visualise the results for the user.

6.2 Project Impact

We expect our system to perform these data mining operations in an efficient and scalable manner. It is important that our application is able to perform these tasks realistically and be an acceptable resource for data miners to use. The result should be a system that validates previous research on algorithm efficiency and scalability, as well as provide a new resource for future data mining work.

6.3 Key Success Factors

The success of our project relies on the following questions:

- Have the chosen algorithms been implemented correctly?
- Does our system work efficiently and scalably on large data sets in a distributed computing environment?
- Does the system provide scalable and useful visualizations of data mining results?

7 PROJECT PLAN

In this section, we will detail a timeline for the project including all planned milestones and deliverables required for the project. We will also identify potential risks that might occur throughout the duration of the project and discuss what resources will be needed for the project.

7.1 Risks

Below is a risk matrix identifying major risks, the effect it might have on the project, and a mitigation plan.

Risk	Impact	Probability	Mitigation
Unable to meet project deadlines	High	Low	Start early, and stick to intermediate deadlines
Twitter data given is too noisy	Medium	Medium	Extra steps will have to be implemented to pre-process and clean the data
Unable to meet system requirements	High	Low	Consult supervisor on problematic areas in early iterations to discuss issues and challenges faced
UCT's HPC cluster is down for the duration of the project	Medium	Low	Apply for another cloud computing service to evaluate the system with

Table 2. Risk matrix

7.2 Timeline

Appendix A includes a Gantt chart of project milestones and deliverables that will be followed for the duration of the project.

7.3 Resources

For the purposes of dealing with large data, a cloud computing platform would be needed to run the parallel MapReduce association rule algorithm. UCT's HPC cluster consists of 24 computers; the majority of these have 4 CPUs with 16 cores per CPU, which is sufficient for the purposes of the project.

In terms of software, we will make use of Apache's Hadoop library to implement the parallelized IR of Twitter data and a distributed implementation of the Apriori algorithm for association rule mining over MapReduce.

7.4 Deliverables

The major deliverables to be handed in are presented in the table below.

Deliverables	Dates
Literature Reviews	4th May 2018
Project Proposal	22nd May 2018
Project Paper	6th September 2018
Project Code	7th September 2018
Poster	19th September 2018
Web Page	26th September 2018

Table 2. Major deliverables and their hand-in dates

These are all marked on the timeline in appendix A.

7.5 Milestones

Highlighted in the Gantt chart presented in Appendix A, the system development phase is planned such that iterative development is used. The first iteration was decided to be the longest as it runs through the university vacation period which gives us time to develop the core functionality. The second iteration has been allocated less time as most core functionality would have already been developed and only improvements would be made in this iteration. Iteration 1 spans from the 11th of June 2018 to the 3rd of August 2018 and iteration 2 from the 6th of August to the 17th of August. Both iterations include a Requirements/Design phase, an Implementation phase, and an Evaluation phase. The first iteration has been assigned more time

since it will include all members first studying up on important libraries and developing a suitable framework for the code. After the code is implemented and evaluated on the second iteration, the final report will be worked on where the results of the evaluating the system will be discussed. The poster, website and project reflection will then be worked on concurrently to finish off the project.

7.6 Work Allocation

Matthew Young will be handling the Information Retrieval step of the project. The retrieved data will then be processed by Eric Dai who will be handling the implementation of the rule mining algorithm. The mined rules will then be visualized by Pieter van der Walt, who is handling the front-end of the software. The front-end is a web application that can also be used to schedule jobs on the system.

REFERENCES

[1] Pritam Gundecha and Huan Liu. 2014. Mining Social Media: A Brief Introduction. In INFORMS Tutorials in Operations Research. Published online: 14 Oct 2014; 1-17. DOI: https://doi.org/10.1287/educ.1120.0105

[2] A.L.Sayeth Saabith, Elankovan Sundararaj, and Azuraliza Abu Bakar. 2016. Parallel implementation of Apriori algorithms on the Hadoop-Mapreduce platform- an evaluation of literature . Journal of Theoretical and Applied Information Technology, vol 85, no. 3 (31 March 2016), pp. 321-351.

[3] Owen de Kretser, Alistair Moffat, Tim Shimmin, Justin Zobel. Methodologies for Distributed Information Retrieval. University of Melbourne, Parkville, Australia

[4] Y. Wang and Y. W. Zhao. 2014. Transplantation of Data Mining Algorithms to Cloud Computing Platform When Dealing Big Data, 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Shanghai, 2014, pp. 175-178. doi: 10.1109/CyberC.2014.39

[5] Olivera Grljević, Zita Bošnjak, Renata Mekovec. 2012. Privacy Preservation in Social Network Analysis. In Central European Conference on Information and Intelligent Systems. September 19-21, 2012. pp 313-318

[6] Solove, DJ. Understanding privacy, Harvard University Press, 2008.

[7] Farzanyar, Z. and Cercone, N., 2013, August. Efficient mining of frequent itemsets in social network data based on MapReduce framework. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 1183-1188). ACM. DOI: http://doi.org/10.1145/2492517.2500301

[8] Lu Zhang, Zhiang Wu, Zhan Bu, Ye Jiang, Jie Cao. 2017. A pattern-based topic detection and analysis system on Chinese tweets, Journal of Computational Science, 2017, ISSN 1877-7503 DOI: https://doi.org/10.1016/j.jocs.2017.08.016.

[9] Agrawal, R., Imieliński, T. and Swami, A., 1993, June. Mining association rules between sets of items in large databases. In Acm sigmod record (Vol. 22, No. 2, pp. 207-216). ACM. DOI: http://doi.org /10.1145/170035.170072

[10] Larena Hoeber, Laura Wood, Ryan Snelgrove, Isabella Hugel, and Dayne Wagner. 2013. Visual twitter analytics: Exploring fan and organizer sentiment during Le Tour de France. In Proceedings of the IEEE VIS Workshop on Sports Data Visualization. IEEE.

[11] twitter.com. (2018). Available at: https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json [Accessed 18 May 2018].

[12] Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), pp.107-113. DOI: http://doi.org/10.1145/1327452.1327492

[13] Rai Krikorian. 2010. Twitter by the numbers. (September 2010). Accessed 19 April 2018 from https://www.slideshare.net/raikrikorian/twitter-by-the-numbers/12- 8_TBper_dayin_total_100

[14] Michael Hahsler and Radoslaw Karpienko. 2016. Visualizing association rules in hierarchical groups. Journal of Business Economics 87, 3 (July 2016), 317–335. DOI:http://dx.doi.org/10.1007/s11573-016-0822-8

[15] Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), pp.107-113. DOI: http://doi.org/10.1145/1327452.1327492

APPENDIX A – PROJECT GANTT CHART