Data Mining and Visualisation of Twitter Using Topic Modelling

Dana Khartabil
H00115090

August 2013

Computer Science
School of Mathematical and Computer Sciences

Dissertation submitted as part of the requirements for the award of the degree of MSc in Computer System Management
Data Mining and Visualisation of Twitter Using Topic Model Modelling

Dana Khartabil

SUPERVISOR: PROF. MIKE CHANTLER

August 15, 2013
## Contents

Declaration .............................................................................................................................................. 6  
Acknowledgment ..................................................................................................................................... 7  
Abstract .................................................................................................................................................. 8  
Chapter 1: Introduction .......................................................................................................................... 9  
  1.1 Aim & Objectives ............................................................................................................................ 10  
  1.2 Requirements Analysis .................................................................................................................... 10  
  1.3 Professional, legal, and ethical issues ............................................................................................... 11  
Chapter 2: Literature Review ................................................................................................................ 13  
  2.1 Topic Modelling .............................................................................................................................. 13  
    2.1.1 Latent Dirichlet Allocation (LDA) ............................................................................................. 13  
    2.1.2 Author-LDA ................................................................................................................................ 14  
    2.1.3 L-LDA ....................................................................................................................................... 15  
    2.1.4 Using Topic Model with Social Networks .................................................................................. 16  
  2.2 Visualisation .................................................................................................................................... 18  
    2.2.1 Word Cloud and Word Storm ...................................................................................................... 18  
    2.2.2 ThemeRiver ............................................................................................................................... 21  
    2.2.3 Navigation ............................................................................................................................... 22  
    2.2.4 TopicNets .................................................................................................................................. 23  
    2.2.5 TwitterScope ............................................................................................................................. 24  
    2.2.6 STREAMIT ............................................................................................................................... 26  
    2.2.7 TopicFlow .................................................................................................................................. 26  
    2.2.8 Termite ..................................................................................................................................... 27  
    2.2.9 Twaphic ..................................................................................................................................... 28  
  2.3 Summary .......................................................................................................................................... 29  
Chapter 3: Data Mining and Visualisation of Twitter Using Topic Modelling ...................................... 30  
  3.1 System Model .................................................................................................................................... 30  
    3.1.1 Dataset ..................................................................................................................................... 31  
    3.1.2 Data Processing ......................................................................................................................... 33
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 1</td>
<td>Graphical model of LDA</td>
<td>14</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>Graphical model of author-LDA</td>
<td>15</td>
</tr>
<tr>
<td>Fig. 3</td>
<td>Graphical model of L-LDA</td>
<td>15</td>
</tr>
<tr>
<td>Fig. 4</td>
<td>Comparison of NMI of the proposed scheme</td>
<td>17</td>
</tr>
<tr>
<td>Fig. 5</td>
<td>Word cloud of twitter user tweets</td>
<td>19</td>
</tr>
<tr>
<td>Fig. 6</td>
<td>Five word clouds</td>
<td>19</td>
</tr>
<tr>
<td>Fig. 7</td>
<td>Coordinated storm visualising six Scientific Programmes</td>
<td>20</td>
</tr>
<tr>
<td>Fig. 8</td>
<td>Trend songs in last.fm</td>
<td>21</td>
</tr>
<tr>
<td>Fig. 9</td>
<td>(a) - (e) Navigating Wikipedia with a topic model</td>
<td>22</td>
</tr>
<tr>
<td>Fig. 10</td>
<td>Topics visualisations using TopicNets (corpus view)</td>
<td>23</td>
</tr>
<tr>
<td>Fig. 11</td>
<td>Topics visualisations using TopicNets (document view)</td>
<td>24</td>
</tr>
<tr>
<td>Fig. 12</td>
<td>(a), (b) Twitterscope visualisation on sports on 16 March 2013</td>
<td>25</td>
</tr>
<tr>
<td>Fig. 13</td>
<td>STREAMIT visualisation on Barack Obama news</td>
<td>26</td>
</tr>
<tr>
<td>Fig. 14</td>
<td>TopicFlow visualisation on US presidential elections debates</td>
<td>27</td>
</tr>
<tr>
<td>Fig. 15</td>
<td>Termite visualisation tool</td>
<td>28</td>
</tr>
<tr>
<td>Fig. 16</td>
<td>Twahpic visualisation tool</td>
<td>29</td>
</tr>
<tr>
<td>Fig. 17</td>
<td>System model for visualising trends in Twitter using topic models</td>
<td>30</td>
</tr>
<tr>
<td>Fig. 18</td>
<td>Hungarian algorithm</td>
<td>35</td>
</tr>
<tr>
<td>Fig. 19</td>
<td>WordStorm of 3 documents about “Obama”</td>
<td>37</td>
</tr>
<tr>
<td>Fig. 20</td>
<td>Word storm visualising tweets about “Syria”</td>
<td>43</td>
</tr>
<tr>
<td>Fig. 21</td>
<td>Word storm visualising tweets about “music”</td>
<td>43</td>
</tr>
<tr>
<td>Fig. 22</td>
<td>Bar chart of Questionnaire answers in terms of LDA and L-LDA clarity</td>
<td>48</td>
</tr>
</tbody>
</table>
Declaration

I “Dana Khartabil” confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: ..............................................

Date: ....................................................
Acknowledgment

I would like to express my appreciation to my supervisor Prof. Mike Chantler for his advices, suggestions and guidance that he offered to me. I am grateful to him for the time he dedicated and effort he put to make this work better.

In addition, I would like to thank my second supervisor, Dr. John Kanyaru, for his constructive comments and useful feedback on my literature review. I used his suggestions in improving my dissertation.

Finally, I would like to thank my family and friends for all their support and encouragement they have offered.
Abstract

Micro-blogging websites have become popular platforms for social media users to communicate and share ideas, news, photos, videos, and all kind of digital information. With such richness of data and wide diversity of users’ types, these websites can provide valuable indicators of current trends, public reaction for breaking news, and interests in emerging topics. Therefore, it is highly important to develop proper tools that can extract and then present topics of interests.

In this project, we adopt Twitter as a social media website to be studied due to its wide popularity, the challenges presented by its unique and limited size messages, and most importantly, the availability of its public streams for research and development purposes. After collecting large sets of data using Twitter streaming API at different periods, data mining, represented by the promising topic modelling methods, is implemented to explore emerging trends in Twitter. Afterwards, we deploy a visualisation tool that can present the abstract output of topic models in a user-friendly way. This tool is able to clearly visualise the selected topics and shows their changes over the time.
Chapter 1: Introduction

Twitter has just celebrated its seventh birthday with amazing statistics that put it in an elite group of social websites. The astonishing statistics include 200 million active users (10 million in the UK) with an average of 400 million tweets per day. Twitter’s wide popularity made it a valuable source of information for a wide and diverse spectrum of users. With Twitter, users have access to real-time information like breaking news, fashion trends, latest scientific discoveries, latest sport scores, and many other kind of information. In other words, tweets can be considered as indications of trends once they occur. Studying and analysing tweets in Twitter becomes important for exploring fashion, social, market, and political trends, e.g., an extensive dataset of tweets in 2012 American election campaigns have been analysed to measure voters reactions to the candidates’ speeches in different US states.

Analysing such a huge set of data needs powerful data tools that can understand Twitter messages and therefore provide us with the emerging trends. Standard text mining and topic modelling algorithms have demonstrated remarkable capabilities of discovering topics in a huge set of documents [1]-[3]. However, these tools cannot be applied to their full potential on Twitter because of the limited length of tweets that cannot exceed 140 characters. In the literature, researchers proposed several techniques to overcome this problem [4]-[8]. A brief summary of those techniques and their effectiveness upon being applied on short messages like tweets will be presented in this report.

Following the discovery of the emerging trends, we need to present this information properly. A natural approach is visualising where the importance of the topics can be presented as graphs, charts, figures, or interactive texts rather than abstract texts or statistics. Several documents visualisation methods are proposed in the literature [9]-[21]. Being designed to visualise huge set of documents rather than social media websites, many of those methods are incapable of tracking the changes in the topics over the time [11]. Only few methods were proposed with the time evolution feature and a very few of those were tested on micro-blogging websites. This has motivated us to develop a visualisation tool to visualise the emerging topics in Twitter and track their changes over the time by applying proper data mining and topic modelling techniques on datasets of tweets collected at different periods.
The rest of the report is organised as follows. The literature review that outlines topic modelling methods and summarises several visualisation techniques is presented in Chapter 2. The system model that represents the main steps of this experiment and the experimental results are introduced in Chapter 3. Chapter 4 focuses on the evaluation of the achieved results. Finally, a conclusion is drawn in Chapter 5.

1.1 Aim & Objectives

The main aim of this project is to develop an application that can visualise trends in Twitter topics and track their changes over the time. Exploring trends in Twitter will be performed using topic models (statistical machine learning approaches), then the uncovered trends will be graphically presented in a way that demonstrate their changes over the time. To achieve this aim, the following objectives will be carried out,

1) Apply topic modelling to short texts like tweets in Twitter.
2) Analyse different datasets collected from Twitter on different days to track the change of trends over the time.
3) Investigate the changes of the topics with time using proper tool(s). The selected tool(s) can be applied on the output of the topic model to show how a certain topic has changed during a period of time.
4) Implement topic visualisation tools that can help users for better understanding of the current trends and their evolution with time.
5) Evaluate the designed visualisation tool using proper evaluation methods.

1.2 Requirements Analysis

In the following, a list of requirements will be provided.

1. Setup dataset of tweets: the first step in this project is to prepare the dataset of the tweets we are planning to analyse using topic models and then visualise. Twitter offers access to small portion of tweets through twitter streaming API for research and development purposes. Twitter recommended using apache commons HTTP client to connect to the stream and collect samples of tweets. Therefore, an application that implements the aforementioned technique will be developed.
2. **Identify trends in twitter topics**: proper topic modelling techniques will be applied on the collected datasets to identify emerging trends in Twitter. A summary of probabilistic topic models is presented earlier in this report. It is crucial that the selected models are capable of handling the short length of tweets. To evaluate the performance of these topic models, a questionnaire will be conducted and the collected data will be analysed using statistical tests.

3. **Implement a visualisation tool**: topic models output is not readable. Therefore, visualising the obtained output in a clear and informative way is highly important. In this project, we will deploy a tool that visually presents the trend topics in Twitter.

4. **Investigate the time evolution of the topics using visualisation tools**: to monitor the time evolution of the topics, we need to develop a visualisation tool that shows the interrelationships between the topics over the time. The functionality of this tool will be evaluated by interviewing some participants to measure how much the tool is readable, easy to understand, and user-friendly.

5. **Final report**: A report that thoroughly explained the techniques and methods used in conducting the project and highlight the main contributions will be delivered. Readers should be able to reach similar results by following the same methods and applying the provided parameters and codes.

### 1.3 Professional, legal, and ethical issues

The project will be conducted professionally, by first choosing a proper topic modelling algorithm that can be applied on short messages like Twitter tweets. The applied topic models should be able to tackle the problem of the limited length of such messages to extract the trend topics. The better the topic model we choose, the faster and more efficient is the moving to the next step of visualising the extracted topics. The chosen topic model will be applied on a pre-collected and refined dataset of tweets. It is essential to investigate the different methods of collecting this data and choose a suitable one. With the enormous number of tweets every day, even 1% of tweets samples for around six hours will have a huge size that may reach a few GBs. Besides, the unique structure of the tweet itself, e.g., containing swears, spams, and useless information as well as special characters, making the process of refining and cleaning the collected tweets a challenging task. After obtaining the trend topics at different time sets, we need to develop a visualisation tool that can be easy to use/understand and yet effective in
Chapter 1: Introduction

showing the evolution of the trends/topics over the time. Finally, some practical tests will be carried out and some important features will be decided like the presentation of the information and the possibility of adding additional features. Correspondingly, the developed tools will be tuned to guarantee a better performance.

The project aims to apply a visualisation tool to visually present the trend topics in social networks, e.g., Twitter. The tool should also track the change of interest in the extracted topics with the time. The dataset will be collected using tools approved and authorised by Twitter, the LDA algorithms and their tools are available for public, and the visualisation will be implemented using licenced software with proper citation. Therefore, there is nothing in the project may be considered as illegal or breaching any law.

The project does not contain any hate acting or violent human rights, as it mainly aim to help users to better understand and track the current trends in Twitter. In addition, all the information presented in this report are cited and referenced in a proper academic way. Besides, any help in the project will be stated and well-acknowledged. Same concept will be applied on the final dissertation presented at the end of the project.

There are arguments about allowing researchers to collect data from the social media websites and concerns of violating privacy policy for the users. Twitter provides legal documents which put some restrictions for the users and researchers to use the twitter data collected by API or other tools [22]. These documents encourage researchers for developments without compromising users' and Twitter's rights. They do not only explain how to access to the twitter’s data but also how the collected data can be used.

Users should be conscious and responsible about all the information they provide throughout this platform and other micro-blogging websites. This includes both general information provided in everyday tweets and personal information. The user who makes his account public has no legal right to complaint about the data being collected from these kinds of sites according to the terms and conditions he/she agreed on upon setting up his/her account*

* [https://twitter.com/tos](https://twitter.com/tos)
Chapter 2: Literature Review

2.1 Topic Modelling

Nowadays, webpages, social networks, academic websites, and news webpages are stacked with enormous set of articles, images, books, videos and different kinds of digital resources are stored. This makes searching and browsing for specific topic among this huge digital cloud are extremely difficult despite the help provided by search engines. For example, Google books include at least 130 million books. First natural question that comes to our mind is what these millions of books contain inside and how can we extract information out of them. There is no human power that can explore and browse through all these information and give precious description of them. Therefore, we need powerful tools that help us search and organise these large set of data, and look at how these topics change with time.

Researchers are working on developing algorithms that can discover hidden thematic structure and organise large sets of documents by exploring the main topics inside them. Those algorithms are called topic modelling.

By definition, Topic models algorithms are statistic methods that detect main topics included in a large set of document and highlight the relation between these topics [1].

The basic assumption of modelling topic is that a document is a collection of words that can be gathered to form topics [1]-[3]. In the following subsection, we will introduce some probabilistic topic models in brief to clarify the main concept of them and highlight the reason of choosing a certain method in extracting topics from Twitter. For deep understanding of topic models, readers are encouraged to check the cited references in the following section.

2.1.1 Latent Dirichlet Allocation (LDA)

LDA is one of the most famous and widely used topic models [2]. It is mainly based on the assumption that documents contain number of topics while topics consist of number of words. Therefore, each document is treated as a probability distribution over one or more topics and each topic is represented by a probability distribution over a set of words. Topics are anonymous in the beginning and the goal is to extract them from the studied data.

For each document in the collection, LDA will follow a generative process defined as:
1. Pick one topic for each document from the topic distribution.
2. Sample a word from the distribution over the words associated with the chosen topic.
3. This process will be repeated for each word in the document.

Fig. 1 Graphical model of LDA [2]

Fig. 1 shows the graphical model of LDA where each node is a random variable that is labelled based on its role in the process. In a collection of documents, each document will be linked with a distribution $\theta_d$ of $T$ topics. Also each topic is associated with a distribution over the words. Each of the topics and words distributions has Dirichlet prior hyper-parameters denoted as $\alpha$ and $\beta_k$, respectively. For each word $W_{d,n}$ in a document $d$, a topic $Z_{d,n}$ is sampled from the associated distribution $\theta_d$. Consequently, a word $W_{d,n}$ from the distribution associated with topic $Z_{d,n}$ is sampled [3]. This process will be repeated $N$ times where $N$ is the number of the word in the document $d$.

From this model, many other topic models were developed like Author-LDA [4], and Labelled-LDA (L-LDA) [5].

### 2.1.2 Author-LDA

This model is an extension of LDA [4], which depends on the same theories of standard LDA. Here, each word in the document, denoted as $W$ will be linked to two latent variables $x$ and $z$ which represent the author and the topic, respectively. Each author will be associated with a distribution $\theta$ over the $T$ topics and each topic is associated with a distribution $\phi$ over words. The
key difference from standard LDA is that in author-LDA will observe a set of authors and words in each studied document. The graphical model of author-LDA is illustrated in Fig. 2.

![Fig. 2 Graphical model of author-LDA [4]](image)

### 2.1.3 L-LDA

L-LDA is another extension of the standard LDA where the model is using observed labels to discover the topics within the studied document [5]. In other words, labels supervise this model while standard LDA can be considered as an ‘unsupervised’ model. In L-LDA, the topic distribution $\theta$ within a document is associated with the topic prior $\alpha$ and the document label set $\Lambda$ which is called an ‘observable variable’. This model is widely used in Twitter where hashtags, emoticons, and social signals can be considered as labels. Fig. 3 shows the graphical model of L-LDA and the relations between model parameters.

![Fig. 3 Graphical model of L-LDA [5]](image)
2.1.4 Using Topic Model with Social Networks

Nowadays many people prefer to connect by using social networks e.g., Facebook, Twitter, Flickr... etc. It is the easiest way to meet new people, communicate with others over the world, share ideas with the people who have similar interests, and discuss different topics, i.e., sports, fashion, trends, news... etc. Recent statistics show how popular such websites are becoming. For example, in June 2010 more than 65 million tweets per day has been reported in Twitter, and the proportion of the visitor were jumped form 6 million in 2009 to 32 million per month in 2011. In October 2012, Facebook celebrated 1 billion users with over a trillion “Likes”, around 140 billion friendship connections, and more than 200bn shared pictures since its launch. With this massive increase of data in net and the rapid increment of users who use social networks. We need powerful tools that enable us organising and searching such large amount of datasets. In our study, we will focus on Twitter as it has become an increasing popular destination for social networking users.

Many researchers try to use topic modelling and standard data mining tools to understand Twitter messages, but they have to overcome the challenges raised by the unique nature of Twitter [6]. First, while topic modelling applies to large corpuses which contain the millions of documents, twitter messages, i.e., tweets, are limited to 140 characters. In addition, users of twitter frequently use some symbol like hashtags (#) to define topics or events and (@) to tag other users, moreover users can use shortened URLs when posting external links. Therefore, from length perspective, tweets may have rich meanings despite their short length. On the other hand, many tweets do not include any useful information or can be spam which we need to avoid them. In [6], a comparison between Twitter and traditional media has been conducted to explore the uniqueness of the information provided in Twitter. Authors chose New York Times (NYT) to represent traditional news media. To explore the topics presented in NYT, standard LDA model has been applied directly on NYT dataset. However, this model cannot be applied on Twitter as tweets are short. To cope with this problem, a Twitter-LDA model has been proposed [6]. The functionality of the proposed model has been evaluated by using standard LDA model where each tweet is considered as a single document and by using author-LDA where all the tweets of a user are considered as a single document. Evaluation results demonstrated that Twitter-LDA model showed better performance than the other two models in terms of exploring topics in Twitter.
LDA and author-LDA models have been applied on Twitter in [7]. To overcome the problem of the short length of tweets, authors proposed several methods to train the selected topic models in Twitter. All the proposed methods depend on aggregating the tweets but with different approaches. While MSG scheme aggregates all training messages, i.e., tweets, which belong to a single user, USER scheme aggregates user profiles and TERM scheme aggregates messages that share pre-defined terms. To evaluate the proposed methods, they have been compared with the author-LDA, denoted as AT, at different aspects. Fig. 4 shows a comparison between the aforementioned models in terms of the normalised mutual information (NMI) between the obtained topics and pre-defined set of topics (based on our knowledge of the categories of all the messages in the dataset). The figure shows that the topic model trained from the collected messages by the same user, i.e., MSG scheme, has outperformed the others.

![Fig. 4 Comparison of NMI of the proposed scheme](image)

In [8], a real time searching engine for twitter using topic modelling was presented. To tackle the problem of applying topic modelling on tweets, authors proposed a process called *supertweets generation*. Tweets that share a group of common features will be aggregated to create a supertweet. Each one of the selected features has been given a weight to prioritise them. As an example, hashtags have been allocated with the highest priority with weight of 1.5 while usernames and retweets were considered as less important with weights of 0.8 and 0.9, respectively. On the other hand, unlike conventional document sets analysed by topic models, topics in tweets evolve, emerge and vanished over time [8]. Therefore, hierarchical LDA were applied on supertweets to extract topics in [8].
2.2 Visualisation

Data visualisation is a way of describing data using graphs, charts, or other kind of diagrams that help viewers to analyse and understand the data quicker and easier. Because of the numerous number of texts available online, there is an urgent need for net explorers to scan large number of text documents in a quick way. A sensible approach is to use visualisation technique to represent the contents of those documents. Along with the importance of the selected visualisation technique, it is also important to choose the proper method of selecting the data to be represented visually. As we mentioned earlier, topic modelling is a favourable tool to explore trend topics in social networks. Visualisation helps us estimating the amount of interest in these trend topics and furthermore enables us tracking this interest over the time.

In literature, several ways of visualising documents have been proposed. In the following, we will describe the state-of-the-art of visualising datasets especially the ones generated by topic models.

2.2.1 Word Cloud and Word Storm

One of the popular and widely used tools for visualising documents is called “word clouds” [9]. It is a graphical tool that represents documents by displaying most frequent word in a document with various groups of font, colour, and size. The size of the word presents the frequency of the word within the text. Fig. 5 shows an example of a cloud of tweets of a twitter user between Nov 2012 and Jan 2013 that contains the top 40 words using http://tweetcloud.icodeforlove.com/.

However, word clouds are incapable of exploring group of documents like the ones in blogs or websites. The main problem is that word clouds are hard to be compared visually. For example, by applying this method on two documents that contain similar topic(s), we will have two different clouds. The generated clouds are difficult to compare as the position of the words are different. In addition, word clouds do not show how the topics vary throughout the time and hence they have been referred to as static word clouds.
As a result, two different approaches were proposed to overcome the aforementioned deficiency of static word clouds, namely, the dynamic word clouds [10] and word storms [11]. Unlike static word clouds, dynamic word clouds illustrate the evolutions of topics of set of documents over the time. It enables users to overview the varying trends in the clouds visually. Dynamic word clouds incorporate geometry-based method in generating word clouds to insure their semantic concurrent and space stability over the time [10]. Fig. 6 demonstrates the generation of dynamic word clouds to track the variation of interest in Apple Inc. over the last decade [10].
This method has two main advantages, its dual-layer visualisation illustrates content evolution at time in detail and its time-variant word cloud layout ensures space efficiency and word location stability. In this way, users can easily observe content changes throughout the generated clouds.

Another approach to cope with the deficiencies of word clouds is word storm [11]. It is a group of word clouds where each one presents one document or describes all the documents in corpus during a period of time. The most important feature of this algorithm is that words which are displayed in several clouds will retain their locations. In this case, it will be easy for the user to point out the difference between the analysed documents.

![Fig. 7 Coordinated storm visualising six Scientific Programmes](image)

Fig. 7 shows an example of applying word storm on six research programmes from five of which are subjects of material sciences while the sixth one represents mathematical sciences programme. We can easily notice that the word “material” was the key expressions in the material sciences programmes while it has been disappeared in the mathematical one. The word “development” has almost the same interest in the six studied programmes.

The key difference between word storm and dynamic word clouds is the fact that maintaining semantic relations between different words within a cloud is different from coordinating
resemblances across clouds, and consequently similar documents do not necessarily being represented by similar clouds.

### 2.2.2 ThemeRiver

It is a visualisation method that represents thematic variations of topics within a large collection of documents over time [12]. It gets its name from using river imagery to represent the theme changes over time. The “river” flows from left to right with time, changing width to represent the changes in the thematic strength of corresponding documents. Each topic will be presented by a coloured current within the river. The width of a current changes with time and indicates the importance of an individual topic (or group of topics) in the analysed set of documents. The output of this method is a river consists of the coloured currents with a corresponding textual description of external events. Fig. 8 shows an example of applying ThemeRiver on trend songs in last.fm [12].

![Fig. 8 Trend songs in last.fm [source: flickr.com]](image_url)

ThemeRiver has been used in [13] to identify topical trends in social media with topic modelling. LDA has been applied on Twitter using ParallelTopics tool [13]. The results of “music” topic with accordance to the BET awards and “flu-like symptoms” topic were presented to demonstrate the usefulness and validation of the proposed approach.
2.2.3 Navigation

Authors of [14] tried to visualise large amount of data in a way that any naive user can handle them and not only an expert of machine learning. To achieve this target, they used a probabilistic topic model, i.e., LDA, to summarise the corpus, and then they displayed the relation between the topic and articles, and between articles and articles. The latter relation leads to an interaction in the visualisation [14]. The proposed navigator can be found at http://bit.ly/wiki100 where it has been applied on Wikipedia and the resulting theme is illustrated in the following figure. Starting from Fig. 9 (a), we can easily notice a set of topics that each of them has been discovered by using topic modelling and represented by a theme. By clicking on the topic “{system, computer, user}”, we move to Fig. 9 (b) where the documents related to the selected topics are listed. The “Operating System” document has been selected and a new window showing the content of this document along with two lists, i.e., related documents and related topics as in Fig. 9(c) . Randomly, we picked the related topic “{math, number, function}” as in Fig. 9(d) and then the related document “Fourier transform” in Fig. 9 (e).

Fig. 9 (a) - (e) Navigating Wikipedia with a topic model

This method is only useful in navigating enormous set of documents like Wikipedia. Besides, it does not show the variation of the topics over the time and therefore, it cannot be applied on our investigation of trend topics in social media.
2.2.4 TopicNets

TopicNets is a web-based system that provides interactive analysis of large sets of data with the help of statistical topic models [15]. It provides corpus and document specific views that we will call them by the web name “TopicNets”. Fig. 10 shows a graph resulted from analysing more than 10,000 research articles from 6 different scientific fields using a statistical topic model, i.e., LDA.

![Fig. 10 Topics visualisations using TopicNets (corpus view)](image)

In this method, documents and topics are located in a node-like graph. Node location is selected based on topic similarity to create visual cluster of documents that share similar topics. Similar to the navigation method of visualising topics, TopicNets allow users to explore the contents of a certain document by clicking on the corresponding *dot* in the corpus figure [15]. By selecting a document from the computer science cluster, we get a document-specific view that reveals the sections related to specific topics as shown in Fig. 11.
Interactivity is not the only common thing between TopicNets and Navigation. Both techniques are most applicable on large corpus with no interest of tracking the changes of the documents over the time.

### 2.2.5 TwitterScope

TwitterScope is a tool that offers real-time monitoring and dynamic visualisation [16]. It monitors the Twitter message stream, pushing new items to the visual foreground. It uses LDA to discover related items and then visual them as countries using a geographic map metaphor. Tweets are treated as towns and cities, where similar messages are close. Then, clusters of highly related messages form countries, each enclosed by a boundary and assigned by a different colour from its surrounding neighbours. The presented layout is dynamic, i.e., it changes over the time to simulate the increase/decrease interest in visualised topics. For an example, we used this tool to analysis tweets related to sport on Saturday 16th March 2013. At 17:10 GMT, TwitterScope generated the map illustrated in Fig. 12 (a). It shows variety of topics and trends each recognised by a colour. In this example we are mainly interested in the one in the middle of the map with the orange colour which has keywords as \{rugby, England, Wales, \ldots\} as the six nations final of rugby between England and Wales was just started. Fig. 12 (b) shows a snapshot of the map at 18:44 GMT where the aforementioned match is concluded making a considerable trend in tweets by Twitter users.
This promising method of visualising is still limited to pre-defined topics and needs to be extended so it can be applied on topics selected by user. In addition, its computational
complexity needs to be further investigated to make it faster and smoother in reconstruction when different time points are chosen.

2.2.6 STREAMIT

STREAMIT is an interactive visualisation system that enables users to explore text streams and monitor their changes over the time [17]. It uses LDA to explore the topics of text streams and dynamic clustering function to allow users to visually recognise clusters of documents with common topics. During visualisation, STREAMIT checks if an incoming document contains one of the extracted topics based on its keywords. As a result, with the time evolution, new clusters may be generated, and formed clustered may grow, split, merge, and/or disappear. Fig.13 demonstrates the functionality of STREAMIT by applying it on news related to Barack Obama on 2010 where the keywords are recognised by colours as follows. “Politics” represented by green, “International Relations” coloured in red, “Terrorism” in yellow, and “Defence and Military” presented by blue.

![Fig. 13 STREAMIT visualisation on Barack Obama news. (a) Aug 2010, 136 articles; (b) after increasing importance of “International Relations”; (c) Sep. 2010, 230 articles [17].](image)

2.2.7 TopicFlow

TopicFlow is a visualisation tool used for exploring popular topics in Twitter and observing their changes over the time [18]. To identify the topics in Twitter data, LDA has been implemented on a large set of tweets. First, the data has been “sliced” in time and LDA has been applied on each time bin to discover the topics during this period. Then, in order to track the changes on the
Chapter 2: Literature Review

topics over the time, a cosine similarity metric has been used to calculate the similarity between topics in different time pins. Cosine similarity compares the probability of the words between topics at different time bins where higher probability of the shared words indicates more similarity between the corresponding topics. Based on this metric, TopicFlow can identify different status of topics, namely, emerging, ending, continuing, and standalone. Each status has been recognised by a different colour as illustrated in Fig. 14 where the size of the node indicates the popularity of the topic. The topic will be considered in its ending state, when it does not appear in the following time bin, while standalone topics are the ones who do not have any relation with other topics in the previous or next time bin. TopicFlow also provides additional features like details of a selected topic, e.g., most probable words and tweets related to this topic. Fig. 14 shows an example of TopicFlow graphical presentation of topics related to the United States presidential elections in 2012.

Fig. 14 TopicFlow visualisation on US presidential elections debates [18].

2.2.8 Termite

Termite is a visualisation technique to assess topic models [19]. It uses LDA as topic model to extract topics from documents and then present the resulted information in a tabular layout as Fig. 15 shows. It offers flexible presenting of topics/terms. Topics can be ordered either by topic index produced by LDA or by topic size while terms can be ordered alphabetically, by frequency
or using seriation [19]. When a topic is selected in the term-topic matrix (Topic 17 in Fig. 15), the system visualises the word frequency distribution relative to the full corpus (bar chart in Fig. 15). Although the tool in its current design allows users to easily identify coherent and significant topics, but yet it does not enable them monitoring the changes of these topics over the time in case of time-varying topics, such as data streams of social platforms.

Fig. 15 Termite visualisation tool [19]. On the left, term-topic distributions are displayed. The bar chart on the right presents the probability of each term

2.2.9 Twaphic

"Twahpic" is a Twitter topic modelling visualisation tool sponsored by Microsoft to show emerging trends in tweets on Twitter in terms of both topics and set of axes, namely, substance, social, status, and style [20]. It uses LDA to identify 200 topics used on Twitter and word clouds with horizontal orientation to visualise these topics. Fig. 16 shows the output of the Twaphic for tweets collected in 30 July 2013.
Fig. 16 Twahpic visualisation tool

However, temporal dynamic that shows the changes of the topics over the time is yet to be implemented.

Other methods of visualising text streams using topics extracting methods other than LDA can be found in [21].

2.3 Summary

This chapter presented a survey of topic modelling techniques applied on micro-blogging websites especially Twitter to discover the current trends and emerging topics. It showed that the small size of Twitter messages limits the effectiveness of standard topic models and therefore some extension models or newly-developed ones were necessary. The state-of-the-art of topics visualisation methods proposed in the literature has been presented and the advantages and disadvantages of each of these methods have been highlighted. Also it showed the significance of the project as many of the visualisation methods in the literature have failed to track how these topics change with time.
Chapter 3: Data Mining and Visualisation of Twitter Using Topic Modelling

In this chapter, a novel application designed to graphically present trends uncovered from Twitter topics using topic models is presented. We will start with proposing the system model and provide a theoretical background of its main components. Afterwards, the functionality of this application from input (collecting dataset) to output (plotting graphics) will be explained in details supported by code and figures (when necessary).

3.1 System Model

To satisfy the main requirements of this project, our system should be able to:

- Collect tweets from Twitter at different time periods (Dataset)
- Prepare the collected dataset to be processed by topic models (Data Processing)
- Identify emerging trends in Twitter topics by deploying proper data mining techniques and prepare the uncovered topics to be visualised (Topic processing)
- Visualise the unveiled topics so user can easily identify the evolving trends in Twitter and monitor their changes over the time (Visualisation tool: WordStorm)

Fig. 17 illustrates the proposed system model. The following subsections will introduce the concepts of the aforementioned system model components.

Fig. 17 System model for visualising trends in Twitter using topic models
3.1.1 Dataset

The dataset we are going to use for our study will be collected from the famous social media website “Twitter” as many of the other social media platforms put restrictions on collecting data from their websites [23]. Twitter offers three ways to collect public data flowing through it using API streaming†. Once applications establish a connection to a streaming endpoint, they are delivered a feed of Tweets. The endpoints can be connected through are:

- **GET statuses/sample**
  It returns a small random sample, i.e., around 1%, of all public tweets on the real time. The tweets returned by the default access level are the same, and therefore clients connected to this endpoint will see the same tweets.

- **POST statuses/filter**
  It returns public tweets that match one or more filter inputs. Multiple parameters may be specified which allows most clients to use a single connection to the Streaming API. At this endpoint, both GET and POST requests are supported, however GET requests with too many parameters may cause the request to be rejected for excessive URL length. In this case, it is recommended to use a POST request to avoid long URLs. To use POST status/filter, we have to specify at least one of the predicate parameters, namely, follow, locations, or track. As we have used this endpoint to collect the tweets of the dataset for this project, we will briefly introduce these predicate parameters.

  - **follow**: specifies the users to return status for in the stream by using a comma separated list of user IDs. In this case, all the tweets related to these users (create, retweet, replied to tweet …….) will be returned by the stream. For example, to follow the tweets issued by BBC in Twitter, [https://stream.twitter.com/1.1/statuses/filter.json?follow=BBC](https://stream.twitter.com/1.1/statuses/filter.json?follow=BBC)

  - **track**: a comma-separated list of phrases or keywords, indicating the keywords to track regardless the text case. E.g., the following request will returns all the tweets that include the term “Syria” [https://stream.twitter.com/1.1/statuses/filter.json?track=syria](https://stream.twitter.com/1.1/statuses/filter.json?track=syria)

† [https://dev.twitter.com/docs/api/1.1](https://dev.twitter.com/docs/api/1.1)
location: a set of bounding boxes, defining geometric locations, to filter Tweets by. Bounding boxes are specified by a comma-separated list of longitude, latitude pairs. As an example,

https://stream.twitter.com/1.1/statuses/filter.json?locations=-2.5,53,-2.5,53.5,-3.4,55,-2.9,56

would match any tweets coming from Manchester OR Edinburgh.

The track, follow, and locations fields should be considered to be combined with an OR operator. For example, track=health&follow=clientX returns tweets matching "health" OR created by user clientX.

– GET statuses/firehose

This endpoint requires special permission to access. It returns all public tweets without restrictions. A creative use of a combination of other resources and various access levels can satisfy nearly every application use case. Table 1 summarises the features of those three methods.

Table 1 Twitter Streaming API Endpoints

<table>
<thead>
<tr>
<th>Resource</th>
<th>GET statuses/sample</th>
<th>POST statuses/filter</th>
<th>GET statuses/firehose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Limited?</td>
<td>Yes†</td>
<td>Yes‡</td>
<td>Yes§</td>
</tr>
<tr>
<td>Authentication</td>
<td>Requires user context</td>
<td>Requires user context</td>
<td>Requires user context</td>
</tr>
<tr>
<td>Response Formats</td>
<td>JSON</td>
<td>JSON</td>
<td>JSON</td>
</tr>
<tr>
<td>HTTP Methods</td>
<td>GET &amp; POST</td>
<td>GET &amp; POST</td>
<td>GET &amp; POST</td>
</tr>
<tr>
<td>Response Object</td>
<td>Tweets</td>
<td>Tweets</td>
<td>Tweets</td>
</tr>
<tr>
<td>API Version</td>
<td>v1.1</td>
<td>v1.1</td>
<td>v1.1</td>
</tr>
</tbody>
</table>

The API allows several types of streams such as:

1) Public streams: provides streams of the public data, i.e., tweets, flowing through Twitter. They are suitable for tracking specific users, trends, or topics, and data mining. They can be collected using the three endpoints we present earlier in this section, GET statuses/sample, POST statuses/filter, and GET statuses/firehose. In this project, we collected Public streams using the endpoint POST statuses/filter.

† 1% of the public stream
‡ The rate is limited based on the client/application access level.
2) *User streams*: offers streams of data related to a specific user. It is useful to track this user’s view of Twitter. The endpoint of these streams is GET user.

3) *Site streams*: can be considered as the multi-user version of user streams. Site streams are mainly used by servers that connect to Twitter on behalf of many users. It can be collected by connecting to the endpoint GET site.

**Hashtag**

The symbol (#), called hashtag, is widely used in Twitter to mark the topics in tweets. It is an important communication tool in Twitter that helps users in finding related tweets and facilities interactivity on the platform. Adding a hashtag to a tweet is similar to joining a group of users in discussing a certain topic [24]. Considering their wide popularity, hashtags are used by Twitter to monitor the trending topics and evolving interests among users.

### 3.1.2 Data Processing

Processing the dataset is highly important and essential for the success of visualising trends in microblogs. Due to the nature of these websites, users can post anything in anyway. Therefore, tweets may contain swears, slangs, hyperlinks, smiley faces, punctuations, and symbols, and many other expressions that can be consider as “noise” and had to be removed. In addition, the format of the tweets collected through the streaming API of Twitter contains a lot of information as can be easily noticed from the tweet sample presented in Appendix C. Based on the application, some parts of this information can be out of the interest of the researchers, and therefore deleting these parts will be beneficial and time-saving for the next steps of the system model. Moreover, in this step, we will overcome the problem of implementing topic models on small corpus like individual tweets by forming a “super tweet” that contains all the tweets collected in a given time period. The procedures we have implemented in processing our dataset will be presented in details later in this report.

### 3.1.3 Topic Processing

#### 3.1.3.1 Topic models

The contents of microblogs can be either theme related, e.g., sports or music, or user reactions to events taking place at a certain time instant, e.g., royal wedding. Uncovering these interests require powerful tools able to characterise the contents of such platforms and overcome
emerging challenges such as the huge scale of data and unique nature of contents. To accomplish this task, we selected two topic models from the literature, namely LDA and L-LDA. Most of similar work conducted in the literature relied on LDA as previously discussed in the literature. Considering the unique nature of tweets in terms of labelling key terms using hashtags (#), we will use also use L-LDA to measure public interests by extracting trends from the collected datasets. The implementation of LDA and L-LDA will be explained in details in discussing experimental results.

### 3.1.3.2 Jensen–Shannon (JS) divergence

JS divergence is a method that generally be used to measure the similarity between two probability distributions [25]. This method depends on common tool called Kullback-Leibler (KL) divergence that measures natural distance for probability distribution A to probability distribution B [26]. So the JS is the known as the average of KL divergence for each distribution, so the KL divergence between two variables A and B can be expressed as:

$$D_{JS} = \frac{1}{2} D_{KL}(A||R) + \frac{1}{2} D_{KL}(B||R)$$

where

$$R = \frac{1}{2} (A + B)$$

$D_{KL}(X||Y)$ denotes the KL divergence between variables X and Y and can be calculated as:

$$D_{KL}(X||Y) = \sum_{i=1}^{M} X_i \log \frac{X_i}{Y_i}$$

where $X_i$ and $Y_i$ are the probability of the i term in the distribution (topic) X and Y, respectively, and $M$ is the number of the terms.

The use of JS divergence is common in topic modelling. In [7], JS divergence was used to measure the similarity between topics generated by different topic modelling schemes. Authors of [23] have used JS divergence to compare the output of topic models applied on datasets collected from Twitter using streaming API and Twitter’s Firehose. Here, we will use JS divergence to measure the similarity between topics generated at different time stamps, i.e., different word clouds in the word storm.
3.1.3.3 Hungarian Algorithm

The Hungarian method is an algorithm which solves assignment problems, i.e., the maximum weighted bipartite matching problems, in polynomial time. Hungarian algorithm was proposed by Kuhn in [27] and reviewed later on by Munkres [28] to reduce its time complexity. In [29], the algorithm has been further developed and a dynamic Hungarian algorithm was proposed to solve the assignment problem in situations with changing costs or weights. To illustrate the functionality of Hungarian algorithm, we will present the steps this algorithm use to solve a given assignment problem. Assume that the weights listed in the Fig. 18(a) represent JS-divergence values between two sets of topics (topics 1-6 vs. topics A-F) that we will apply Hungarian algorithm on. In practice, the dimensions of a matrix of divergence values between two topics or the same topic at different time instant can be high. Therefore, using Hungarian algorithm to match the paired topics is indispensible.

<table>
<thead>
<tr>
<th></th>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
<th>Topic 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic A</td>
<td>0.07</td>
<td>0.22</td>
<td>0.19</td>
<td>0.21</td>
<td>0.31</td>
<td>0.19</td>
</tr>
<tr>
<td>Topic B</td>
<td>0.16</td>
<td>0.20</td>
<td>0.10</td>
<td>0.13</td>
<td>0.09</td>
<td>0.28</td>
</tr>
<tr>
<td>Topic C</td>
<td>0.24</td>
<td>0.29</td>
<td>0.12</td>
<td>0.13</td>
<td>0.08</td>
<td>0.11</td>
</tr>
<tr>
<td>Topic D</td>
<td>0.13</td>
<td>0.23</td>
<td>0.15</td>
<td>0.10</td>
<td>0.24</td>
<td>0.09</td>
</tr>
<tr>
<td>Topic E</td>
<td>0.29</td>
<td>0.36</td>
<td>0.17</td>
<td>0.16</td>
<td>0.17</td>
<td>0.09</td>
</tr>
<tr>
<td>Topic F</td>
<td>0.22</td>
<td>0.30</td>
<td>0.15</td>
<td>0.08</td>
<td>0.06</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Fig.18 Hungarian algorithm (a) input values (b) output values (recognised by grey shades)
The steps of this algorithm are summaries as follows:

**Step 1**: *Find the minimum for each row and subtract it from all the entries.*
As a result, each row has at least one zero.

**Step 2**: *Find the minimum for each column and subtract it from all the entries in the column.*
Consequently, each row and each column has at least one zero.

**Step 3**: *Draw lines across each row and column in way that you covered all zeros, by using the minimum number of the lines.*

**Step 4**: *If the line you drawn is n which is the number the rows in the matrix so we achieve what we need. If not, proceed to the next step.*

**Step 5**: *Find the smallest entry which is not covered by any of the lines, and subtract it from each entry which is not covered by the lines. Then, add it to each entry which is covered by a vertical and a horizontal line. Get back to step 3.*

The output of the Hungarian algorithm that shows the paired topics in our example is presented in Fig. 18(b).

### 3.1.4 Visualisation (Word storm)

As we mentioned earlier in this report, word storm is a visualisation tool which avoids the deficiencies encountered in the famous word clouds in analysing and comparing corpus of documents [11]. The complexity of this algorithm resulted from maintaining the same word in almost the same place in sequential clouds. We have chosen this tool as it satisfies the demand of users/researchers of an easy way to scan a large number of documents, check the shared subjects among them, and enables users to track the changes of these subjects at different time periods.

The words encountered in different documents, or documents with updated versions, will appear in the same location with same colour and orientation in the clouds. In this case, the comparison between the different clouds representing different documents will be efficient and easy. We can easily see the words appear/disappear, getting larger/smaller in the clouds which indicates the relative importance of these words. Fig.19 presents a word storm of three documents about US president “Obama”. The clouds of the storm present common words with same colour and different size depending on the percentage of words repetition in the document.
Word storm code is available to download from developers’ website**. Word storm mainly depends on WordCram which is an open source library††. Generating a WordCram follows the next steps. First, words are weighted by how many times they appear in a document. The library offers some additional attributes for the user to choose from, such as case sensitivity (lowercase, uppercase, and keep words as default), removing numbers from the corpus, and deleting common English words, i.e., stop words, or a list of words specified by the users. Then, users can choose to the style of words within the clouds, e.g., size, angle, font, colour...etc. Finally, users can draw WordCram based on their selected attributes. WordCram generally helps in generating individual clouds while Word storm is generated by sharing the layout of these clouds. It coordinates the colour, orientation, and the position of the words shared between different clouds. This can be performed by implementing a novel algorithm that combines iterative and gradient algorithms [11]. Here, we will introduce the aforementioned algorithms in brief.

- **Iterative layout algorithm**: that iteratively produces word clouds by changing the position of the shared words to place them in the same location in all clouds. However, in practice the algorithm pull words away from centre as this makes it easier to locate common words in same location in different clouds. This results in unsightly sparse layouts with many white spaces.

**http://groups.inf.ed.ac.uk/cup/wordstorm/wordstorm.html#code
††http://wordcram.org
Chapter 3: Data Mining and Visualisation of Twitter Using Topic Modelling

- **Gradient algorithm**: allows us to avoid hideous sparse layouts that might be generated by the iterative algorithm. However, it requires several iterations to converge and the final layout mainly depends on the initialisation.

- **Combined Algorithm**: combines the previous two algorithms using the layout in the output of the iterative algorithm as an input of the gradient one and therefore improves the initial layout significantly. The gradient method drags words closer to the centre to create a more compact layout. In addition, it tends to pull together the locations of shared words that the iterative method failed to converge to a single position.

3.2 Experimental Results

3.2.1 Dataset

The dataset has been collected from the Twitter using API streaming. We used predicate parameters, i.e., track and language to filter the stream of tweets. For an example, [https://stream.twitter.com/1.1/statuses/filter.json?track=samsung&language=en](https://stream.twitter.com/1.1/statuses/filter.json?track=samsung&language=en) will only collect tweets about “Samsung” in English. For our project we have initially picked up “Syria” as a subject as it has been a hot topic which draws a lot of attention over the last two years. People from all over the world showed a lot of interest in the news related to Syria on a daily basis and they have used social media platforms, including Twitter, to raise concerns or discuss events which resulted in a large sets of data... In order to choose another trend or evolving topic in Twitter, we have collected the tweets of four different subjects represent different areas of interests, i.e., music, health, burger (representing food), and Samsung (referring to technology). In a week time, we noticed that “music” has the highest number of tweets with a considerable margin in compare with the other topics which agrees with the observation reported in [30]. Table 2 summarises the average number of tweets per day for the aforementioned subjects.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Tweets per Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syria</td>
<td>85,689</td>
</tr>
<tr>
<td>Music</td>
<td>1,127,117</td>
</tr>
<tr>
<td>Samsung</td>
<td>18,143</td>
</tr>
<tr>
<td>Burger</td>
<td>39,190</td>
</tr>
<tr>
<td>Health</td>
<td>68,317</td>
</tr>
</tbody>
</table>

Table 2 Examples of average number of tweets per day for different topics
Chapter 3: Data Mining and Visualisation of Twitter Using Topic Modelling

It worth to be noted that the selection of analysing tweets related to “syria” is not based on the number of tweets only, but also considers the concise of the related words in compare with the sparse and wide variety of expressions noticed in other topics such as “health”.

**JSON**

JSON (JavaScript Object Notation) is a text format that is easy to read and write for human. It uses JavaScript syntax to describe data objects, but it is still language and platform independent**‡‡**. An example of the syntax of JSON can be found in Appendix C.

The tweet is full of important information provided friendly by the JSON format. It includes data about the user proceeded by @ character, when the tweet has posted, label which is proceeded by # character and the tweet text. Based on this format, we need to process the JSON files to extract the contents we are interested in, i.e., label, tweet text and time stamp.

**3.2.2 Data processing**

In this step, the collected dataset will be prepared to be processed by the topic models in the next phase.

1). All the links and stop words (such as the, a, an, don’t, have, has …) included in tweet texts will be removed. These words are not helpful in topic mining and considered as obstacles in processing the data as they usually exist in high frequency without providing useful information. In addition, all the tweets that are less than 3 words have a weak semantic weight and therefore will be considered as insignificant and will be removed as well.

2). For L-LDA, we will use hashtag (#) to indicate labels of the tweets [23]. The importance and the popularity of hashtags have been stated earlier. In this case, tweets with no hashtags will be removed from the dataset.

3). Change the text case into lower case for all the labels, for example (#Syria will be converted to #syria) as they are referring to the same topic.

4). Tweets with labels that have considerably low number of frequency, and therefore low rank of significance, will be removed as well, e.g., labels encountered 100 times in a corpus of

**‡‡**[http://www.json.org/](http://www.json.org/)
620500 tweets. This has significantly improved the presentation of the data and saved the
time of the topic model process as such tweets can be considered as a noise in the corpus.
Applying the last four steps will reduce the noise in corpuses and make the semantic relation
between the terms more powerful.
These steps of filtering are essential for preparing the dataset for topic models by discarding
irrelevant information and maintaining valuable ones. Number of pre-processed tweets can
be 9572 filtered from 64000 pre-processed tweets. The output of this step is an excel file with
tweets information separated into columns, e.g., the format of the excel file for LLDA,

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
</table>

Where Y: year, Mo: month, D: day, H: hour, Mi: minutes, S: seconds

### 3.2.3 Topics Processing

1). Apply both LDA and L-LDA on the corpus resulted from the previous steps.

**Topic modelling:** We have used the Stanford Topic Modelling Toolbox (TMT) provided by
Stanford Natural Language Processing (NLP) Group. The tool has been made available for
researchers and scientist to analyse textual data§§. TMT offers several topic models, namely,
LDA, L-LDA, and Partially Labelled Dirichlet Allocation (PLDA) [31].
The tool will read the tweets text from the excel file that contains the filtered and pre-
prepared dataset, i.e., column 10 in our case as shown in the example below.
Then, it will remove all the punctuations in the text and split up the input text by whitespace
characters or specify a custom tokenize using regular expression. The number of distinct
words can be reduced by making all the characters lowercase, CaseFolder(), similar to what
we have done earlier for the tweets labels. Afterwards, Non-number and non-word characters
are removed from the generated lists of tokenized. In addition, this tool gives the user the
choice to remove all English stop words by using StopWordFilter("en"). Additional stop
words can be defined manually using TermStopListFilter.
In L-LDA, the number of topics is the number of unique labels. For fair comparison and
without compromising the accuracy of our work, the number of topics in LDA was set to be

§§ [http://nlp.stanford.edu/software/tmt/tmt-0.4/](http://nlp.stanford.edu/software/tmt/tmt-0.4/)
equal to the number of topics, i.e., labels, of L-LDA for a given corpus. The most important output files of topic modelling are:

**00000, 00050… 01000:** Snapshots of the model during training. In our project, we used the output of the iteration 01000.

**Params.txt:** contains model parameters used during training

**Summary.txt:** summary of the topic model with top 20 terms per topic and how many words instance of each have occurred.

**Term-index txt:** mapping from terms in the corpus to ID number

**Topic-term distributions.csv.gz:** from each topic, the probability of each term in that topic.

2). After applying LDA and L-LDA on different corpuses at different time spans, we used JS-divergence to measure the similarity between the resulted topics in different time stamps. The program is written in Java and the code will be presented in the Appendix A. JS-divergence will read terms’ distributions and indexes from term distribution and term index files, respectively, from the output of the TMT. The value of the JS-divergence between each topic in the first corpus and the topics in the second corpus, that represents another time span, will be listed in an excel file. The smaller the value the closer the topics are.

3). Hungarian method will be applied on the output of the JS-divergence to match each topic at certain time instant with a suitable topic in the next time instant based on the corresponding JS-divergence value. In other words, it will help us to find the similarity between topics at different time spans.
Chapter 3: Data Mining and Visualisation of Twitter Using Topic Modelling

The table below presents two corpus of topic “music” at different time periods. Both topics are associated with the same label “jazz” and therefore a similarity in their terms can be easily noticed. However, we can also notice that some words observed in 15 July 2013 have been disappeared in 20 July 2013 and replaced with new terms.

<table>
<thead>
<tr>
<th>15 July 2013</th>
<th>20 July 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>#jazz</td>
<td>179959.0</td>
</tr>
<tr>
<td>playing</td>
<td>2552.0</td>
</tr>
<tr>
<td>jazz</td>
<td>2550.0</td>
</tr>
<tr>
<td>Luis</td>
<td>1558.0</td>
</tr>
<tr>
<td>live</td>
<td>1384.0</td>
</tr>
<tr>
<td>tune</td>
<td>1133.0</td>
</tr>
<tr>
<td>duke</td>
<td>910.0</td>
</tr>
<tr>
<td>ellington</td>
<td>898.0</td>
</tr>
<tr>
<td>youtube</td>
<td>876.0</td>
</tr>
<tr>
<td>missing</td>
<td>871.0</td>
</tr>
<tr>
<td>people</td>
<td>860.0</td>
</tr>
<tr>
<td>artists</td>
<td>859.0</td>
</tr>
<tr>
<td>listen</td>
<td>846.0</td>
</tr>
<tr>
<td>blues</td>
<td>707.0</td>
</tr>
<tr>
<td>playback</td>
<td>672.0</td>
</tr>
<tr>
<td>world</td>
<td>662.0</td>
</tr>
<tr>
<td>collaboration</td>
<td>647.0</td>
</tr>
<tr>
<td>scene</td>
<td>625.0</td>
</tr>
<tr>
<td>band</td>
<td>592.0</td>
</tr>
<tr>
<td>think</td>
<td>583.0</td>
</tr>
<tr>
<td>songs</td>
<td>569.0</td>
</tr>
<tr>
<td>#jazz</td>
<td>122091.0</td>
</tr>
<tr>
<td>Luis</td>
<td>1734.0</td>
</tr>
<tr>
<td>jazz</td>
<td>1729.0</td>
</tr>
<tr>
<td>live</td>
<td>1372.0</td>
</tr>
<tr>
<td>listen</td>
<td>808.0</td>
</tr>
<tr>
<td>songs</td>
<td>762.0</td>
</tr>
<tr>
<td>people</td>
<td>671.0</td>
</tr>
<tr>
<td>playback</td>
<td>624.0</td>
</tr>
<tr>
<td>playing</td>
<td>609.0</td>
</tr>
<tr>
<td>life</td>
<td>608.0</td>
</tr>
<tr>
<td>friends</td>
<td>592.0</td>
</tr>
<tr>
<td>heart</td>
<td>543.0</td>
</tr>
<tr>
<td>band</td>
<td>512.0</td>
</tr>
<tr>
<td>enjoy</td>
<td>485.0</td>
</tr>
<tr>
<td>join</td>
<td>482.00</td>
</tr>
<tr>
<td>beat</td>
<td>456.0</td>
</tr>
<tr>
<td>check</td>
<td>438.0</td>
</tr>
<tr>
<td>sound</td>
<td>430.0</td>
</tr>
<tr>
<td>dance</td>
<td>421.0</td>
</tr>
<tr>
<td>tune</td>
<td>412.0</td>
</tr>
<tr>
<td>artists</td>
<td>380.0</td>
</tr>
</tbody>
</table>

3.2.4 Visualisation Tool (Word storm)

To visualise the emerging topics in Twitter and their change over the time, we are going to use Word storm proposed in [11]. The input files are the collected, filtered, and processed corpus of tweets at different time periods. The word storm will produce the corresponding set of clouds where each cloud represents one of these periods, i.e., trending terms in tweets about “music” in a given day. To fit the word storm in our work, a new class called FileLoaderStorm is introduced to the project. The main functionality of this class is to parse the Stanford TMT algorithms, i.e., LDA and L-LDA, output files or any files contains group of topics and use the same format. This is done by loading all the files located in the input folder considering each file to represent only one storm where storm composed of group of topics and each topic contains a group of weighted words. Figs 20 and 21 show word storms for tweets about “Syria” using LDA and “music” using L-LDA, respectively. Fig. 20 clearly shows that chemical weapons draw a lot of attentions at that period while it can be easily noticed from Fig. 21 that Rihanna, the British singer, is in the spotlight when it comes to music.
Chapter 3: Data Mining and Visualisation of Twitter Using Topic Modelling

Fig. 20 Word storm visualising tweets about “Syria” between 18 May and 8 June 2013

Fig. 21 Word storm visualising tweets about “music” between 17 July and 29 July 2013
3.3 Summary

A system model for data mining and visualisation of Twitter using topic models was proposed. The proposed system consists of four main components, namely dataset, data processing, topic processing, and visualisation tool. We started by introducing background information of the proposed system model components. Several methods of collecting tweets from Twitter to compose datasets was thoroughly explained, followed by presenting different data processing techniques to filter and reshape these datasets and eliminate any noise might be encountered before applying two topic model algorithms, i.e., LDA and L-LDA. In addition, the mechanism of JS-divergence and Hungarian algorithm and their roles in discovering the similarity between the topics was introduced. Then, word storm that represent the visualisation tool, the last component of our system model, was described. The algorithms of the word storm that distinguish it from the popular word cloud were presented. These algorithms enabled us to get a storm (set of clouds) of popular topics in Twitter with preserve terms’ attributes, e.g., location, colour, orientation over the clouds, i.e., different time periods. This technique offers an easy yet informative way for users to track the changes of topics over the time. After providing the comprehensive background description of the proposed system model components, we explained the implementation of those components in our project. We have demonstrated the functionality of our system using datasets collected from Twitter for two different subjects, Syria and music, and processed by two different topic models, LDA and L-LDA, and graphically presented by word storms to show how the terms associated with these two topics are changing with time. Our system will be evaluated in the following chapter using proper evaluation methods.
Chapter 4: Evaluation

In this chapter, the performance of our system will be evaluated from two perspectives as follows.

4.1 Topic Modelling Evaluation

Here we will evaluate and compare between LDA and L-LDA in terms of clarity in representing topics from the dataset collected from social media like Twitter. This is highly important because of the difficulties that topics models face in being applied on such type of data, e.g., limited length of tweets, as we mentioned earlier in the literature review.

The questionnaire takes ten topics randomly picked up from the output of each topic model, i.e., LDA and L-LDA. The participants in the questionnaire have been asked to read these topic and decide how clear they are by choosing of a grade between 1 to 5 where 1 means the topics are very unclear and 5 denotes that topics are very clear. 24 participants were involved in this questionnaire and their answers, summarised in Appendix B, are analysed and studied using proper statistical test as we are going to explain in the following sections.

4.1.1 Statistical tests

Statistical test is a method that helps researchers to make a decision using experimental data. In order to evaluate the two studied topic models, LDA and L-LDA, we have to carefully setup a questionnaire and select the corresponding statistical test. In the literature, there are various types of statistical tests that suit different applications. Choosing a suitable test depends on the case, data, and participants you have. On other words, it depends on the variables that you are going to measure. Therefore, before picking up our test, we need to fully understand the variables that we have. Here, we will briefly define and explain different types of variables. Variables are defined as things that vary between people or can change over the time [32]. As most hypotheses can be explained from cause/effect perspective, related variables can be categorised correspondingly into independent variables (predictor variables) that are considered to be the cause of some effect and dependent variables (outcome variables) which are affected by the any change occurred in the independent ones.

Variables can be also categorised based on the level of measurements which represents the relation between the measured variable(s) and the corresponding numbers, e.g., marking presentation effectiveness on a scale of 1-10. In terms of level of measurements, variables can be
classified as *categorical* and *continuous*. We called variables as categorical when the studied entities can be split into specific categories, e.g., trees can be orange, apple, or oak but cannot be oak and apple at the same time. Three kinds of categorical variables can be recognised [32]:

- **Binary variable**: there are only two distinct categories, e.g., male or female.
- **Nominal variable**: more than two groups are exist, e.g., liquid, gas, or solid.
- **Ordinal variable**: similar to nominal variables but here the categories have a specific order, e.g., lecturer, reader, or professor.

Continuous variables can be distinct by score and they can be further classified into two types:

- **Interval variable**: equal distances on the variable denote equal differences in the things that are measured, e.g., in shoe sizes, the difference between 5” and 6” is similar to the difference between 11” and 12”.
- **Ratio variable**: it is similar to the interval variable, but the scores ratio should be meaningful, e.g., a score of 9 on delicious food scale means that it is triple times better than a meal scores 3.

Based on the variables that we have, a proper statistical test will be selected. The selection criteria and a comprehensive analysis of statistical test results will introduced in the following section.

### 4.1.2 Survey Data Analysis

As we have explained earlier, survey’s participants have been asked to evaluate the clarity, the outcome variable, of the two topic models using a 5-point scale. A variable like seems interval, but the rating of the clarity is solely based on participants’ subjective feeling. In other words, we are not certain that two participants give a topic a same score are found it equally clear. Similarly, we cannot consider that a topic scored 4 is twice as clear as a one scored 2. Therefore, the outcome variable is regarded as ordinal [32].

Considering that we are comparing between two categories, LDA and L-LDA, with the help of the same group of participants and using the clarity (ordinal variable) as the evaluation metric, the Wilcoxon signed-rank test is selected. It is a non-parametric test used to evaluate a single sample, or compare between two samples. This test is very similar to Mann-Whitney test except that the participants in the latter one should be in different groups for the two samples [32].
The results of the Wilcoxon signed rank test are presented in Tables 3, 4 and 5. Table 3 presents descriptive statistics such as the mean value, standard deviations, minimum, and maximum of the scores. Table 4 includes information about the ranks. Negative ranks, that represents the case of LDA is better than the L-LDA, are only 2 while positive ranks, denoting that L-LDA is clearer than LDA, are 18. The equality between both models, represented by ties, has been encountered 4 times out of the total 24 results. This table also shows the average and the sum of positive and negative ranks. The footnotes below this table explain to what the positive and negative ranks associate. The smaller value of the sum of the ranks is called the test statistic and denoted as $T$, and therefore $T = 30$ (sum of negative ranks) in our test. Consequently, Table 5 shows that the test statistics is based on negative ranks, and calculate Z-score and P-value. P-value is a probability number between 0 and 1 that determines how significant the difference between the two studied groups is. The higher the P-value, the more likely it is that the difference was caused by chance. Correspondingly, if the value is less than 0.05, it means that the two samples are significantly different [32]. Z-score is the value of an observation expressed in standard deviation unites. From Table 5, Z-score is -2.919 and, after neglecting the minus sign, is bigger than 1.96 (that represents the most common confidence interval = 95%). It is even bigger than the 99% confidence interval which is equal to 2.58. This indicates that our test is significant at P < 0.05.

To sum up, the Wilcoxon signed-rank test showed that the difference between LDA and L-LDA is significant ($Z = -2.919$, $P < 0.05$, and $T = 30$). As Z-score is calculated based on the negative ranks, it means that L-LDA is significantly clearer than LDA.

Fig. 22 shows bar charts of questionnaire participants’ feedbacks in terms of the two topic models clarity.

<table>
<thead>
<tr>
<th>Table 3 Descriptive Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>LDA</td>
</tr>
<tr>
<td>LLDA</td>
</tr>
</tbody>
</table>
Wilcoxon Signed Ranks Test

Table 4 Ranks

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean Rank</th>
<th>Sum of Ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLDA - LDA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative Ranks</td>
<td>2</td>
<td>15.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Positive Ranks</td>
<td>18</td>
<td>10.00</td>
<td>180.00</td>
</tr>
<tr>
<td>Ties</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a. LLDA < LDA
b. LLDA > LDA
c. LLDA = LDA

Table 5 Test Statistics

<table>
<thead>
<tr>
<th></th>
<th>LLDA – LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>-2.919a</td>
</tr>
<tr>
<td>Asymp. Sig. (2-tailed)</td>
<td>.004</td>
</tr>
</tbody>
</table>

a. Wilcoxon Signed Ranks Test
b. Based on negative ranks.

Fig. 22 Bar chart of Questionnaire answers in terms of LDA and L-LDA clarity

Our results agree with the ones presented in [33] where the performance of applying LDA and L-LDA on datasets extracted from Twitter was evaluated using perplexity and authorship prediction accuracy. Both metrics have showed that L-LDA has significantly outperformed LDA in characterising contents on Twitter.
4.2 Word storm Evaluation

To evaluate the outcome of the software which is a word storm visualising trends in Twitter and their changes over the time, interviews with 10 participants have been conducted. At the beginning of each interview, we briefly introduced the main idea of the tool and explained the functionality and applications of word storm. Then, interviewees have been given the chance to fill in a form to express their ideas about the tool and provide any comments or suggestions. 70% of the participants reported that the tool clear and easy to understand. Although 60% of them described the tool as informative and understandable, moreover, 60% found it easy to track the changes of topics over the time. The rest of the participants, i.e., 40%, struggled to track a word between a set of words over different clouds representing different time periods. Some suggestions to improve the presentation have been raised, like presenting the words which only appear in all or most of the clouds, using dark colours to present common words between the clouds and light ones for less frequent words, or using the horizontal orientation only instead of both horizontal and vertical ones. Many users preferred to have fewer words per cloud for better presentation and easier tracking. Summary of interview sheets are presented in Appendix D.

4.3 Summary

The performance of our proposed system has been evaluated. We have used questionnaires to evaluate the clarity of topics extracted from Twitter datasets and interviews to evaluate the generated word storms. A statistical test, called Wilcoxon signed-rank, has been carefully selected to process the results of the questionnaire. The output of the test has shown that L-LDA has better performance than LDA in extracting topics from set of tweets. The evaluation of the visualisation output showed promising results with 70% of participants was satisfied with the clarity of the generated figures and 60% were able to easily track the changes of the topics over the time.
Chapter 5: Conclusions

In this chapter, a summary of all the work performed in this project is provided and the achieved results, key observations, and contributions are highlighted. Moreover, the difficulties we have encountered during conducting this project are listed. Finally, potential research directions and future developments are outlined.

5.1 Summary and Conclusions

The aim of the project is to track the changes of trend topics in social media websites, like Twitter, over time. The requirements and the professional, legal, and ethical issues related to the project have been discussed at the beginning of this report. It is important to conduct this research in line with these ethical issues due to unique nature of the data, i.e., personal status and messages, we are using. Choosing Twitter was based on the fact that most micro-blogging websites have complete restrictions on accessing users’ data while Twitter offers sample of its public streams to researchers and developers. To characterise the contents of social media websites, powerful data mining tools are needed. Topic models have been selected as they showed promising results in uncovering trends and emerging topics in similar platforms. A survey of these models and the challenges they have to overcome when applied on Twitter has been introduced in Chapter 2. The importance of presenting the extract output of topic models graphically has been demonstrated and recent topics visualisation methods for Twitter contents have been presented and discussed in details. It showed that most of these visualisation methods are incapable of displaying the changes of the trends over the time. Therefore, we proposed a system model that implements topic models to uncover trends in Twitter and present them visually using word storm to show their temporal evolution. The proposed model consists of four main components, dataset, data processing, topic processing, and visualisation tool. Comprehensive background information of each of these components was presented. Then, the implementation of each component in relation with the others was explained in Chapter 3. The functionality of our proposed system was demonstrated using datasets collected from Twitter at different time periods. The generated word storms have showed how topics has changed over the time by producing several word clouds, each represent a certain time instant, and preserve the location of the topics over these clouds. In Chapter 4, an evaluation of our system was presented.
First, we have evaluated the implementation of two topic models, LDA and L-LDA, on datasets collected from Twitter. The evaluation was performed using questionnaire were 24 participants have been asked to rate the clarity of each topic model output. Then, we have used Wilcoxon sign-rank test to process the questionnaire results. It has been shown that L-LDA outperforms LDA in terms of the clarity of the topics or trends extracted form Twitter. This agrees with other results presented in the literature indicating the validity of our results. To evaluate the output of our system, 10 interviews were conducted. 70% of the participants have appreciated the clarity of the visualisation tool and 60% of them have found it easy to track the changes of the trends with the time.

5.2 Difficulties

The main hurdles in this project were the dataset collected from Twitter and the evaluation due to the following reasons.

- **Collecting the tweets:** In order to collect tweets from Twitter we initially used `GET statuses/sample` to connect to the API streaming. Despite the fact that only a 1% sample of public tweets will stream through this connection, this is still a lot of data considering the huge popularity of Twitter ***. For an example, the average number of tweets collected between 17 April 2013 and 15 May 2013 was 7,344,000 (with an average size of 18.28 GB) per day. Therefore, a broadband and staple internet connection with huge storage space was needed. Any fluctuation in the connection means that the streaming will go down and we have to establish another connection. Although we have implemented an auto reconnect function for occasion failures but this may make the situation worse in case of an unstable internet connection as excessive connection attempts, either successful or not, might lead to an automatic ban of the user IP by Twitter. We have faced several situations were API streams stopped flowing causing a temporal gap in the collected data. In addition, establishing a connection with the API streaming through one of the endpoints used to require a basic authentication, i.e., username and password. However, in 11 June 2013, Twitter announced the retirement of basic authentication support on streaming API and

*** According to Twitter, the average number of tweets per day is over 450 million.
replaced it with a more complicated authentication procedure using OAuth\textsuperscript{†††}. Users are now required to create an application, create access token, and then generate an OAuth signature (cURL command) to replace the old basic HTTPs connection.

- **Diverse of topics**: Topics in social media platforms are sparse and widely diverse as users’ status can be about anything and written in any language. Initially, we simply used \url{https://stream.twitter.com/1.1/statuses/sample.json} to collect the tweets without providing any restrictions on language or topics. After collecting over 500 GB of data over 28 days, we started processing this data to find that the topics were inoperative and impractical because of their incoherence and inconsistent. Therefore, we had to start another session of collecting the data. This time we used \texttt{POST statuses/filter} to collect English tweets only and filter the tweets based on selected topic(s), e.g., Syria, Music, and Health. All of the examples provided in this report are based on the data collected using this method.

- **Processing time**: Processing the data and get it prepared so topic models algorithms could be applied is an extensively time-consuming process. Applying topic models on large corpuses, like the ones we have in this project, takes a long time as well. In addition, JS-divergence and Hungarian algorithms are expensive computationally and require a considerable amount of time to be performed. For these reasons, having a good time plan played an important role in the success of the project.

- **Data processing**: Processing the data in the output of the topic models was another challenge we had to face in this project. Irrelevant and uninformative data can be considered as a noise that will negatively impact the clarity and effectiveness of the visualisation tool. Such data can be resulted from slang words, swears, or sometimes abbreviations. This problem has become more obvious in common subjects like music where we had to study the topics in order to remove these terms by including them in the list of stop words. It is worth noting that different subjects may have different stop words and establishing such lists is highly important but time consuming.

- **Evaluation**: In the evaluation of topic models and visualisation tool using questionnaire and interviews, respectively, it was difficult to find a large number of participants. We have

\textsuperscript{†††} OAuth is an authentication protocol that allows users to approve application to act on their behalf without sharing their password.
distributed the questionnaire to over 80 persons but only 24 participants submitted their answers as well as 10 participants were willing to perform interviews to discuss the final output of the project which are word storms of trends in Twitter. However, this does not compromise the accuracy of both evaluations as the above numbers satisfies the requirements of acceptable number of testers, i.e., 5-10, stated in MSc project & dissertation guidelines proposed by MACS in Heriot-Watt University.

5.3 Future Work

Visualising trends and monitor their changes over the time using streams of social media users’ feed is an important topic that has a lot of potential for development. We believe that we can build on the promising results we have achieved in this project and further develop the software. This can be performed by

- Considering the suggestions provided by testers in evaluating the visualisation tool to enhance the visualisation tool in terms of clarity and user-friendly, such as maintaining one orientation of the terms within the cloud and using high contrasted colours to highlight valuable ones. A list of important suggestions has been provided earlier in this report.

- Developing the software to offer flexible options of visualisation tool. For example, users can choose a topic from a provided set of topics, select the period he would like to track the changes of these topics within, and control number of words presented per word cloud. In addition, it will be useful to provide the user with more details about the topics, like adding a side bar to every cloud showing the percentage of each word relative to the other words within the cloud.

- Implementing other topic models like PLDA [31] which is similar to L-LDA but it allows more than one topic per label and offers a set of background labels.

- Further improving the evaluation by designing a survey that compares our tool with other Twitter visualisation tools, such as ThemeRiver and TopicFlow.
References


Appendices

Appendix A: Source Code

Program 1: Data Processing

```java
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.lang.Exception;
import java.lang.String;
import java.util.StringTokenizer;
import org.json.JSONArray;
import java.lang.String;
import java.io.IOException;
import java.lang.Object;
import org.json.JSONObject;

/**
 * This program read the JSON file and take the information we need from the
 * Tweets and remove all links and some characters.
 * The output will be the csv file.
 */
public class ProcessData {
    public static void main(String[] args) {
        try {
            long count = 0;
            long doc = 0;
            String hashCheck = "#";
            String jString = "";
            File dd = new File("data");
            File[] array;
            array = dd.listFiles();
            for (int j = 0; j < (array.length); j++) {
                System.out.println(array[(j)]);
                FileReader f = new FileReader(array[(j)]);
                BufferedReader br = new BufferedReader(f);
                JSONObject currentObject = new JSONObject(jString);
                String tweet = "";
                String labels = "";
                String userName = "";
                count++;
                System.out.println(count);
                System.out.println(array[(j)]);
                if (!currentObject.has("lang")
                    continue;
                String lang = currentObject.getString("lang");
                if (!lang.equals("en")
                    continue;
            }
        }
    }
}
```
if (currentObject.has("user")) {
    JSONObject userObject;
    userObject = currentObject.getJSONObject("user");
    userName = userObject.getString("screen_name");
    labels += "@" + userName;
}

if (currentObject.has("created_at")) {
    String[] dateTimeInfo = currentObject.getString("created_at").split(" ");
    String label = dateTimeInfo[5];
    labels += "," + label;
    label = dateTimeInfo[1] + "-" + label;
    labels += "," + label;
    label = dateTimeInfo[2] + "-" + label;
    labels += "," + label;
    dateTimeInfo[3].split(":");
    label = dateTimeInfo[0] + "-" + label;
    labels += "," + label;
    label = dateTimeInfo[1] + "-" + label;
    labels += "," + label;
    label = dateTimeInfo[2] + "-" + label;
    labels += "," + label;
    String[] timeInfo = dateTimeInfo[3].split(":");
    label = timeInfo[0] + "-" + label;
    labels += "," + label;
    label = timeInfo[1] + "-" + label;
    labels += "," + label;
    label = timeInfo[2] + "-" + label;
    labels += "," + label;
} else {
    continue;
}

if (currentObject.has("text")) {
    tweet = currentObject.getString("text");
    StringTokenizer stk = new StringTokenizer(tweet);
    while (stk.hasMoreTokens()) {
        String nextToken = stk.nextToken();
        char firstCharacter = Character.toString(firstCharacter);
        if (hashCheck.equals(firstCharacterString)) {
            if (firstCharacterString.length() <= 2) {
                if (nextToken.matches("#[a-zA-Z]+")) {
                    nextToken = nextToken.toLowerCase();
                    labels += nextToken + ",";
                }
            } else if (labels.isEmpty() || labels == null)
                || (tweet.isEmpty() || tweet == null)) {
        } else {
            // remove references from tweet
tweet = tweet.replaceAll("\[#&\]\S+", "");

// remove urls from tweet
tweet =
tweet.replaceAll("(http|https|ftp):\S+", "");

// removes some punctuation
tweet = tweet.replace("", "");
tweet = tweet.replace("\n", "");
tweet = tweet.replace("\r", "");
tweet = tweet.replace("\", "");
labels = labels.replace("\", "");
if (tweet.split("\\s").length > 5 &&
labels.contains("#")) {
    String[] label = labels.split("#");
    for (int i = 1; i < label.length; i++) {
        fw.write(doc + "," + label[0] + "+" + label[i] + tweet + 
"");
        doc++;
    }
}
fw.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
    return;
} catch (IOException e) {
    e.printStackTrace();
    return;
} catch (JSONException e) {
    e.printStackTrace();
    return;
}
}
**Program 2: JS-Divergence**

```java
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

// This program to calculate the JS-divergence between the topics.
public class JsDivergence {
    static String currentLine;
    public static void main(String[] args) throws IOException {
        String report = "";
        // read files that contain topic distribution and term index
        String[] term11 = readLines("C:/Users/Dana/Desktop/file1/term-index.txt");
        String[] term22 = readLines("C:/Users/Dana/Desktop/file2/term-index.txt");
        float[][] file1 = readFile("C:/Users/Dana/Desktop/file1/topic-term-distributions.csv");
        float[][] file2 = readFile("C:/Users/Dana/Desktop/file2/topic-term-distributions.csv");
        int[] matchTerm1 = matchTerm(term11, term22);
        int[] matchTerm2 = matchTerm(term22, term11);
        for (int i = 0; i < file1.length; i++) {
            for (int j = 0; j < file2.length; j++) {
                report += jensenShannonDivergence(file1[i], file2[j], matchTerm1, matchTerm2) + ",";
            }
        }
        report += "\n";
        IO.writeToFile("C:/Users/Dana/Desktop/TJS.csv", report);
    }

    // To normalise the the float numbers and call the method KLdivergence
    public static double jensenShannonDivergence(float[] t1, float[] t2, int[] m1, int[] m2) throws IOException {
        int n1, n2;
        n1 = t1.length;
        n2 = t2.length;
        float sum1 = 0, sum2 = 0;
        for (int a = 0; a < n1; a++) {
            sum1 += t1[a];
        }
        for (int b = 0; b < n2; b++) {
            sum2 += t2[b];
        }
        float[] nor1 = new float[n1];
        float[] nor2 = new float[n2];
        for (int a = 0; a < n1; a++) {
            nor1[a] = t1[a] / sum1;
        }
        for (int b = 0; b < n2; b++) {
            nor2[b] = t2[b] / sum2;
        }
        return (klDivergence(nor1, nor2, m1) + klDivergence(nor2, nor1, m2)) / 2;
    }
}
```
To calculate the KL divergence between two numbers:

```java
public static double klDivergence(float[] p1, float[] p2, int[] match) {
    final double log2 = Math.log(2);
    double klDiv = 0.0, P = 0.0;
    for (int i = 0; i < match.length; i++) {
        if (match[i] >= 0) {
            if (p1[i] != 0.0 && p2[match[i]] != 0.0) {
                P = (p1[i] + p2[match[i]]);
                klDiv += p1[i] * Math.log(2 * p1[i] / P) / log2;
            }
        }
    }
    if (klDiv == 0) klDiv = 1;
    return klDiv;
}
```

**The method will go through two array and give the index of the item in the second array which match item in the first array.**

@param t1 array of string items

@param t2 array of string items

@return array of index

```java
public static int[] matchTerm(String[] t1, String[] t2) {
    int[] match = new int[t1.length];
    for (int i = 0; i < t1.length; i++) {
        match[i] = -1;
        for (int j = 0; j < t2.length; j++) {
            if (t1[i].equalsIgnoreCase(t2[j])) {
                match[i] = j;
                break;
            }
        }
    }
    return match;
}
```

// This method to read line from file

```java
public static String[] readLines(String filename) throws IOException {
    FileReader fileReader = new FileReader(filename);
    BufferedReader bufferedReader = new BufferedReader(fileReader);
    List<String> lines = new ArrayList<String>();
    String line = null;
    while ((line = bufferedReader.readLine()) != null) {
        lines.add(line);
    }
    bufferedReader.close();
    return lines.toArray(new String[lines.size()]);
}
```

// This method to read file

```java
@SuppressWarnings("resource")
public static float[][] readFile(String fileName) throws FileNotFoundException, IOException {
    File file = new File(fileName);
    FileReader fr = new FileReader(file);
    BufferedReader br = new BufferedReader(fr);
    int r = 0; // Read the file
    while ((line = br.readLine()) != null) {
        Scanner scanner = new Scanner(line);
```
scanner.useDelimiter("","");
list.add(new ArrayList<String>());
while (scanner.hasNext()) {
    list.get(r).add(scanner.next());
    r++;
}

// Convert the list into an int[][]
float[][] data = new float[list.size()][];
for (int i = 0; i < list.size(); i++) {
    data[i] = new float[list.get(i).size()];
    for (int j = 0; j < list.get(i).size(); j++) {
        data[i][j] = Float.parseFloat(list.get(i).get(j));
    }
}
return data;
Program 3: Label Processing

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;

// This program to process the label for L-LDA
public class Label {
    /**
     * @param args
     * @throws IOException
     */
    @SuppressWarnings("resource")
    public static void main(String[] args) throws IOException {
        String currentLine;
        FileReader fr1 = new FileReader("C:/Users/Dana/Desktop/DATA Process/file.csv");
        BufferedReader br1 = new BufferedReader(fr1);
        HashMap<String, HashSet<String>> labelsTweets = new HashMap<String, HashSet<String>>();
        ArrayList<String> tweets = new ArrayList<String>();
        while ((currentLine = br1.readLine()) != null) {
            tweets.add(currentLine);
            String parts[] = currentLine.split(",");
            String key = parts[8];
            HashSet<String> value = labelsTweets.get(key);
            if (value == null) { value = new HashSet<String>();}
            value.add(parts[9]);
            labelsTweets.put(key, value);
        }
        for (String tweet : tweets) {
            String parts[] = tweet.split(",");
            String key = parts[8];
            int count = labelsTweets.get(key).size();
            if (count > 100) {
                fw.write(tweet + "\n");
            }
        }
    }
}
```
Appendix B: Topic Model Evaluation: Questionnaire

**Topic Models: A Survey**

Please read each topic provided below (which are groups of words) and decide on a scale of 5 how clear and meaningful the topics are where 1 denotes very unclear and 5 means very clear.

* Required

### 1-LDA

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>chemical weapons</td>
<td>israeli</td>
<td>killed</td>
<td>rights</td>
<td>group</td>
</tr>
<tr>
<td>arms</td>
<td>israel</td>
<td>fighters</td>
<td>human</td>
<td>council</td>
</tr>
<tr>
<td>use</td>
<td>fire</td>
<td>near</td>
<td>times</td>
<td>blocks</td>
</tr>
<tr>
<td>iran</td>
<td>after</td>
<td>terrorists</td>
<td>council</td>
<td>security</td>
</tr>
<tr>
<td>rebel</td>
<td>vehicle</td>
<td>were</td>
<td>debate</td>
<td>actions</td>
</tr>
<tr>
<td>used</td>
<td>russia</td>
<td>lebanese</td>
<td>killed</td>
<td>vote</td>
</tr>
<tr>
<td>evidence</td>
<td>refugees</td>
<td>more</td>
<td>resolution</td>
<td>terrorist</td>
</tr>
<tr>
<td>claims</td>
<td>troops</td>
<td>city</td>
<td>new</td>
<td>kerry</td>
</tr>
<tr>
<td>reporting</td>
<td>destroyed</td>
<td>death</td>
<td>chief</td>
<td>military</td>
</tr>
<tr>
<td>lack</td>
<td>gunshots</td>
<td>people</td>
<td>fighter</td>
<td>iran</td>
</tr>
<tr>
<td>investigate</td>
<td>border</td>
<td>today</td>
<td>urgent</td>
<td>terror</td>
</tr>
<tr>
<td>skepticism</td>
<td>airport</td>
<td>least</td>
<td>abbas</td>
<td>peace</td>
</tr>
<tr>
<td>pub</td>
<td>president</td>
<td>far</td>
<td>douma</td>
<td>hezbollah</td>
</tr>
<tr>
<td>reports</td>
<td>chemical</td>
<td>source</td>
<td>bomb</td>
<td>shots</td>
</tr>
<tr>
<td>embargo</td>
<td>international</td>
<td>border</td>
<td>hezbollah</td>
<td>police</td>
</tr>
<tr>
<td>turkish</td>
<td>arrest</td>
<td>soldiers</td>
<td>government</td>
<td>fired</td>
</tr>
<tr>
<td>seen</td>
<td>team</td>
<td>another</td>
<td>children</td>
<td>troops</td>
</tr>
<tr>
<td>russia</td>
<td>military</td>
<td>toll</td>
<td>rebel</td>
<td>fight</td>
</tr>
<tr>
<td>obama</td>
<td>major</td>
<td>clashes</td>
<td>assad's</td>
<td>dies</td>
</tr>
</tbody>
</table>

### Topics 6

<table>
<thead>
<tr>
<th>Topic 7</th>
<th>Topic 8</th>
<th>Topic 9</th>
<th>Topic 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>rebel</td>
<td>shelling</td>
<td>access</td>
<td>fighters</td>
</tr>
<tr>
<td>arm</td>
<td>damascus</td>
<td>security</td>
<td>resistance</td>
</tr>
<tr>
<td>just</td>
<td>rural</td>
<td>rebel</td>
<td>opposition</td>
</tr>
<tr>
<td>groups</td>
<td>homs</td>
<td>aid</td>
<td>world</td>
</tr>
<tr>
<td>resolution</td>
<td>video</td>
<td>humanitarian</td>
<td>foreign</td>
</tr>
<tr>
<td>allies</td>
<td>aleppo</td>
<td>region</td>
<td>soldiers</td>
</tr>
<tr>
<td>fighters</td>
<td>city</td>
<td>council</td>
<td>military</td>
</tr>
<tr>
<td>al Qaeda</td>
<td>deir</td>
<td>final</td>
<td>destroyed</td>
</tr>
<tr>
<td>fund</td>
<td>vet</td>
<td>help</td>
<td>down</td>
</tr>
<tr>
<td>vote</td>
<td>heavy</td>
<td>conflict</td>
<td>town</td>
</tr>
<tr>
<td>council</td>
<td>free</td>
<td>allow</td>
<td>city</td>
</tr>
<tr>
<td>voted</td>
<td>israel</td>
<td>capture</td>
<td>invasion</td>
</tr>
<tr>
<td>clerics</td>
<td>rocket</td>
<td>stronghold</td>
<td>checkpoint</td>
</tr>
<tr>
<td>condemns</td>
<td>people</td>
<td>missing</td>
<td>militia</td>
</tr>
<tr>
<td>urges</td>
<td>evidence</td>
<td>french</td>
<td>liberated</td>
</tr>
<tr>
<td>arab</td>
<td>criminal</td>
<td>qusair</td>
<td>arab</td>
</tr>
<tr>
<td>chief</td>
<td>international</td>
<td>allow</td>
<td>border</td>
</tr>
<tr>
<td>join</td>
<td>arrest</td>
<td>fall</td>
<td>refugees</td>
</tr>
<tr>
<td>end</td>
<td>refugees</td>
<td>been</td>
<td>turkish</td>
</tr>
<tr>
<td>conflict</td>
<td>assad's</td>
<td>qusayr</td>
<td>terms</td>
</tr>
<tr>
<td>call</td>
<td>only</td>
<td>u.n</td>
<td>calls</td>
</tr>
<tr>
<td>united</td>
<td>still</td>
<td>time</td>
<td>human</td>
</tr>
</tbody>
</table>
## LDA

<table>
<thead>
<tr>
<th>Topic</th>
<th>1 (Very Unclear)</th>
<th>2 (Unclear)</th>
<th>3 (Neutral)</th>
<th>4 (Clear)</th>
<th>5 (Very Clear)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic 1</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 2</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 3</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 4</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 5</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 6</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 7</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 8</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 9</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Topic 10</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
## Topic Models: A Survey

### 2-Labeled-LDA

<table>
<thead>
<tr>
<th>Syria</th>
<th>UN</th>
<th>Hezbollah</th>
<th>Iraq</th>
<th>Obama</th>
</tr>
</thead>
<tbody>
<tr>
<td>city</td>
<td>resolution</td>
<td>fighting</td>
<td>iraq</td>
<td>no-fly</td>
</tr>
<tr>
<td>military</td>
<td>assembly</td>
<td>town</td>
<td>state</td>
<td>zone</td>
</tr>
<tr>
<td>town</td>
<td>general</td>
<td>troops</td>
<td>violence</td>
<td>plans</td>
</tr>
<tr>
<td>children</td>
<td>million</td>
<td>lebanese</td>
<td>iraqi</td>
<td>report</td>
</tr>
<tr>
<td>today</td>
<td>council</td>
<td>qusayr</td>
<td>just</td>
<td>plan</td>
</tr>
<tr>
<td>chemical</td>
<td>security</td>
<td>rebel</td>
<td>government</td>
<td>obama</td>
</tr>
<tr>
<td>israeli</td>
<td>refugees</td>
<td>support</td>
<td>don't</td>
<td>action</td>
</tr>
<tr>
<td>like</td>
<td>chief</td>
<td>military</td>
<td>stop</td>
<td>stop</td>
</tr>
<tr>
<td>talks</td>
<td>vote</td>
<td>border</td>
<td>countries</td>
<td>syria's</td>
</tr>
<tr>
<td>attack</td>
<td>torture</td>
<td>terror</td>
<td>islam</td>
<td>rocket</td>
</tr>
<tr>
<td>just</td>
<td>death</td>
<td>city</td>
<td>allah</td>
<td>fears</td>
</tr>
<tr>
<td>obama</td>
<td>condemning</td>
<td>terrorists</td>
<td>report</td>
<td>military</td>
</tr>
<tr>
<td>number</td>
<td>toll</td>
<td>government</td>
<td>policy</td>
<td>victim</td>
</tr>
<tr>
<td>russia</td>
<td>official</td>
<td>killing</td>
<td>bashar</td>
<td>rebel</td>
</tr>
<tr>
<td>fighting</td>
<td>conference</td>
<td>near</td>
<td>night</td>
<td>i raq</td>
</tr>
<tr>
<td>rebel</td>
<td>calls</td>
<td>attacks</td>
<td>bombs</td>
<td>lebanon</td>
</tr>
<tr>
<td>terrorists</td>
<td>push</td>
<td>turkey</td>
<td>imagine</td>
<td>civilians</td>
</tr>
<tr>
<td>report</td>
<td>abuses</td>
<td>border</td>
<td>israeli</td>
<td>home</td>
</tr>
<tr>
<td>conflict</td>
<td>violence</td>
<td>attack</td>
<td>defense</td>
<td>white</td>
</tr>
<tr>
<td>missiles</td>
<td>condemns</td>
<td>fighting</td>
<td>russian</td>
<td>rocket</td>
</tr>
</tbody>
</table>

### Iran

<table>
<thead>
<tr>
<th>Iran</th>
<th>US</th>
<th>Revolution</th>
<th>Erdogan</th>
<th>Humanrights</th>
</tr>
</thead>
<tbody>
<tr>
<td>embassy</td>
<td>attack</td>
<td>truth</td>
<td>protests</td>
<td>rights</td>
</tr>
<tr>
<td>protest</td>
<td>government</td>
<td>terrorists</td>
<td></td>
<td></td>
</tr>
<tr>
<td>civil</td>
<td>change</td>
<td>free</td>
<td></td>
<td></td>
</tr>
<tr>
<td>beirut</td>
<td>london</td>
<td>armed</td>
<td>minister</td>
<td></td>
</tr>
<tr>
<td>iraq</td>
<td>hits</td>
<td>units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iranian</td>
<td>opposition</td>
<td>qusayr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iran</td>
<td>department</td>
<td>shelling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>latest</td>
<td>drills</td>
<td>sniper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>border</td>
<td>ground</td>
<td>damascus</td>
<td></td>
<td></td>
</tr>
<tr>
<td>supports</td>
<td>time</td>
<td>syrians</td>
<td></td>
<td></td>
</tr>
<tr>
<td>funding</td>
<td>spokesperson</td>
<td>government</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sides</td>
<td>terrorists</td>
<td>hezbollah</td>
<td></td>
<td></td>
</tr>
<tr>
<td>playing</td>
<td>country</td>
<td>jordan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>protesters</td>
<td>jordan</td>
<td>missile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>world</td>
<td>border</td>
<td>threatens</td>
<td></td>
<td></td>
</tr>
<tr>
<td>russia</td>
<td>jets</td>
<td>hezbollah</td>
<td></td>
<td></td>
</tr>
<tr>
<td>armed</td>
<td>clear</td>
<td>refugees</td>
<td></td>
<td></td>
</tr>
<tr>
<td>outside</td>
<td>policy</td>
<td>troops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mojahedin</td>
<td>reuters</td>
<td>killing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sectarian</td>
<td>week</td>
<td>dead</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labeled-LDA *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (Very Unclear)</td>
<td>2 (Unclear)</td>
<td>3 (Neutral)</td>
<td>4 (Clear)</td>
<td>5 (Very Clear)</td>
</tr>
<tr>
<td>Syria</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>UN</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Hezbollah</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Iraq</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Obama</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Iran</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>US</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Revolution</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Erdogan</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Humanrights</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

« Back  Submit

Never submit passwords through Google Forms.
Appendix C: Sample of Tweet in JSON file

Appendix D: Visualisation Tool Evaluation: Interview Questions and answers

The figures presented below shows tweets about news taken at different time stamps. We want to evaluate the clarity and effectiveness of this visualisation method in reflecting the changes of topics in Twitter over the time.

Q1: Is this visualisation tool clear and easy to understand? If not, why? For example,

- A lot of words in one storm
- It is difficult to find the topics within an individual word cloud, i.e., at certain time instance.

Your answer goes here

Q2: Do you think that this way is good to present the changes of the topics over the time?
If yes, how to describe this method (Tick all possible answers):

☐ Informative
☐ Understandable
☐ User-friendly
☐ Easy to track a topic change over the time (monitor the topic over the word clouds)

If not, please state why?

Your answer goes here

Q3: If we want to make this visualisation tool works better, what do you suggest? For example,

- Add the percentage of each word (instead of depending on the font size)
- Add details about the words appearance/disappearance,
- Only present the words that exist at different time stamps (i.e., appear in all word clouds)

Your answer goes here

Q4: Do you have any other suggestions or comments?

Your answer goes here
## Interview Answers

| Participant 1 | A1: The figures contain lots of words, in general they are not easy to understand and contain some difficult words  
A2: Informative  
A3: Effective Use some of the words in bright colors, reduce the number of words to increase clarity. The use of words one-Thread gradient colors and away from the other colors of threads  
A4: No |
| Participant 2 | A1: No, because there are a lot of word and in different direction  
A2: No, not understandable  
A3: Less words and in the same direction and some explanation, Different color and size is a good idea  
A4: Write percentage near each word with size change. Same direction |
| Participant 3 | A1: The presentation is clear in term of one cloud, i.e., a specific day, or even few days, e.g., 2-3 days. However, for longer duration, e.g., 6 time spans, it gets hard to visually track a specific topic among the other.  
A2: yes, the method is informative, understandable and user-friendly. It shows new topics emerging and other fainting and even disappearing with the time. The preservation of the color and location of topics in different word clouds has make it easier to track the change of these topics over time. But still, for less important words, it is not that easy to track.  
A3: Using horizontal orientation only could be helpful. It might be useful to choose bright colors to highlight important topics and dim colors to represent less important ones. Adding percentage to each word instead of size might make the cloud more crowded.  
A4: No |
| Participant 4 | A1: No, too many words in one storm...hard to get the point and takes a lot of time.  
A2: No, because it’s complicated way and most of the readers don’t have the time and patience in order to analyze it and track a topic change over time.  
A3: Only present the words that exist at different time stamps or use fewer words  
A4: No |
| Participant 5 | A1: It is clear and easy to understand the main topic for each plot.  
A2: Easy to track topics change over the time and understandable.  
A3: The way of presentation the topics is very efficient. Adding more details will be confused.  
A4: No |
| Participant 6 | A1: It is clear and easy to understand.  
A2: Informative, Understandable, and Easy to track a topic change over the time  
A3: Add details about the words appearance and disappearance.  
A4: Make all the words which appear in the cloud related to the topics. |
<p>| Participant 7 | A1: Yes, it is clear to represent a topic using this visualization tool. At a specific time the topic can be easily understood. Furthermore, preserving the words locations among word clouds make the task easier for the reader to track the... |</p>
<table>
<thead>
<tr>
<th></th>
<th>A1: yes, it is clear and easy to track the topics in each cloud.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A2: informative, understandable, user-friendly and easy to track.</td>
</tr>
<tr>
<td></td>
<td>A3: using horizontal orientation could be helpful to track the changes over time. In addition to give chance for the user to choose the number of the words in the clouds.</td>
</tr>
<tr>
<td></td>
<td>A4: No</td>
</tr>
<tr>
<td><strong>Participant 8</strong></td>
<td></td>
</tr>
</tbody>
</table>

|                  | A1: It is quite easy to understand. However it is just give a general idea about the topics. |
|                  | A2: I think that this way is present in brief details. More details about the topics should be added. However, the changes of the topics with time are clear and easy to track. |
|                  | A3: Add details about the words and topics.                       |
|                  | A4: No                                                            |
| **Participant 9**|                                                                  |

|                  | A1: Clear but NOT VERY EASY and a lot of word                     |
|                  | A2: Understandable                                                 |
|                  | A3: Less words and in the same direction, with different color and size is ok |
|                  | A4: If we can but the word from bigger size to smaller size and with order from up to down and keep the same color that will be more clear |
| **Participant 10**|                                                                 |
Appendix E: Project Plan

Tasks and Timetables

<table>
<thead>
<tr>
<th>Task name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visualisation and data mining using topic modelling</td>
<td>100 days</td>
<td>Mon 08/04/13</td>
<td>Thu 22/08/13</td>
</tr>
<tr>
<td>1. Twitter API</td>
<td>36 days</td>
<td>Mon 08/04/13</td>
<td>Fri 24/05/13</td>
</tr>
<tr>
<td>1.1. Collect data</td>
<td>25 days</td>
<td>Mon 08/04/13</td>
<td>Thu 09/05/13</td>
</tr>
<tr>
<td>1.2. Define relevant tweets</td>
<td>5 days</td>
<td>Fri 10/05/13</td>
<td>Thu 16/05/13</td>
</tr>
<tr>
<td>1.3. Extract tweets</td>
<td>5 days</td>
<td>Fri 17/05/13</td>
<td>Thu 23/05/13</td>
</tr>
<tr>
<td>2. Learning JavaScript</td>
<td>7 days</td>
<td>Fri 17/05/13</td>
<td>Mon 27/05/13</td>
</tr>
<tr>
<td>3. Topic Modelling</td>
<td>18 days</td>
<td>Fri 24/05/13</td>
<td>Tue 18/06/13</td>
</tr>
<tr>
<td>3.1. Prepare the data to be processed by topic modelling</td>
<td>9 days</td>
<td>Fri 24/05/13</td>
<td>Wed 05/06/13</td>
</tr>
<tr>
<td>3.2. Apply Topic model on the collected dataset to extract the topics</td>
<td>9 days</td>
<td>Thu 06/06/13</td>
<td>Tue 18/06/13</td>
</tr>
<tr>
<td>4. Visualisation</td>
<td>27 days</td>
<td>Wed 19/06/13</td>
<td>Thu 25/07/13</td>
</tr>
<tr>
<td>4.1. Decide how to display the results</td>
<td>8 days</td>
<td>Wed 19/06/13</td>
<td>Fri 28/06/13</td>
</tr>
<tr>
<td>4.2. Implementation</td>
<td>19 days</td>
<td>Mon 01/07/13</td>
<td>Thu 25/07/13</td>
</tr>
<tr>
<td>5. Thesis write-up</td>
<td>14 days</td>
<td>Fri 26/07/13</td>
<td>Wed 14/08/13</td>
</tr>
<tr>
<td>6. Deliver the project</td>
<td>0 days</td>
<td>Thu 15/08/13</td>
<td>Thu 15/08/13</td>
</tr>
<tr>
<td>7. Poster of the Project</td>
<td>6 days</td>
<td>Fri 15/08/13</td>
<td>Thu 22/08/13</td>
</tr>
</tbody>
</table>

The Timeline for the tasks is

Start Mon 08/04/13

Finish Thu 22/08/13
The Gantt chart of the project is: