

EURECOM  
Department of Multimedia  
Campus SophiaTech  
CS 50193  
06904 Sophia Antipolis cedex  
FRANCE

Project Title

Collecting microposts shared on social networks and reconstructing timelines  
of Events

June 29<sup>th</sup>, 2014

Chaima Ben Rabah

Supervisor(s): José Luis Redondo Garcia, Raphaël Troncy, Giuseppe Rizzo

Email : {Chaima.Ben-Rabah, Jose-Luis.Redondo-Garcia, raphael.troncy,  
Giuseppe.Rizzo}@eurecom.fr

## Contents

List of figures .....	1
List of tables .....	2
1	Introduction
2	Background and related works
3	Overview of the Topic detection methods
3.1	Introduction
3.2	Latent Dirichlet Allocation
3.3	Document-pivot topic detection
3.4	Graph-based feature-pivot topic detection
3.5	Soft frequent pattern mining
3.6	BNgram
4	Popularity
4.1.	Description
4.2.	Results and Evaluation
5	Topic detection framework
5.1.	Methodology
5.2.	Preparing the DataSet
5.2.1	Dataset creation
5.2.2	Difficulties faced
5.3	Implementation
5.4	Evaluation and results
6	Popularity Method
6.1	Description

6.2	Results and evaluation
7	Topic evaluator
7.1	Implementation
7.2	Evaluation and Results
8	Topic Viewer
8.1	Features supported and input data
8.2	System Structure Design
8.3	Software Architecture
8.4	Topic Viewer implementation
8.4.1	Tweets Collection Module
8.4.2	Keywords Extraction Module
8.4.3	Topic visualisation module
8.5	System evaluation and conclusion
9	Discussion and Conclusion
	References

## List of figures

<b>Figure 1:</b> Twitter activity during FACup event .....	21
<b>Figure 2:</b> Curve of the distribution of the tweets (according to their popularity).....	12
<b>Figure 3:</b> Application background process for collecting tweets.....	12
<b>Figure 4:</b> Application background process for extracting topics.....	16
<b>Figure 5:</b> Internal working of the apache Tomcat box.....	20
<b>Figure 6:</b> SIMILE timeline .....	20
<b>Figure 7:</b> Distribution of tweets per day for the Hashtag 'copaAmerica' .....	20
<b>Figure 8:</b> Twitter analysis of 'copaAmerica' 20 minutes before the collection.....	20
<b>Figure 8:</b> Topic Viewer interface .....	20

## List of Tables

<b>Table 1:</b> Results automatically detected by LDA .....	14
<b>Table 2:</b> Results automatically detected by Doc-p .....	15
<b>Table 3:</b> Results automatically detected by BNgram .....	14
<b>Table 4:</b> Results automatically detected by SFIM .....	15
<b>Table 5:</b> Results automatically detected by Popularity .....	14
<b>Table 6:</b> Comparison of Topic Detection Algorithms .....	14
<b>Table 7:</b> Comparison of the metrics for different values of @N.....	15

# 1 Introduction

The number of online social networks is increasing allowing internet users to share their life events, activities, photos and many other contents. Large quantities of information are shared everyday through social networks, making them attractive sources of data for social network research and also for journalist.

To detect relevant topics and events from this huge amount of data, we can use a variety of methods that we will explore in this report. The report will be mainly divided into two parts: The first part will contain an overall test and evaluation of six of these methods, and the second part will contain an implementation and evaluation of a Java project developed for the purpose of collection of tweets and extraction of topics from it.

In this first part, we have two main objectives : extract newsworthy topics for an event for a given time interval and use an evaluation script that, based on a ground truth datasets that has been manually selected from a mainstream news sources, can compute a list of performance measures.

This report is organized as follows: we present the context and some related works in Section 2. We will be presenting the process of preparing the database which will be used with the different topic detection techniques presented in Section 3 as a necessary initial step; Section 4 is dedicated to describe the Topic detection framework used to detect the topics from the selected dataset; A novel technique for topic extraction was proposed in Section 5. Section 6 describes how we use the Topic evaluator to evaluate the result of the five topic detection methods in a second place. Subsequently, section 7 is reserved to the Topic Viewer, a Java project developed to implement the Topic detection framework and another tool for collecting tweets directly from Twitter API. We finish in Section 8 with a brief discussion and conclusion.

## 2 Background and Related Work

Keywords are a set of words presenting the overall topic of the document. Keyword extraction refers to how to extract certain words or phrases from a document automatically to present the document's topic objectively and accurately. It has become a basis of several text mining applications such as topic detection.

There are proposed topic detection techniques for the extraction of keywords from large amounts of online text like tweet's text. They largely fall in three classes: The first, termed document-pivot methods, group together individual documents according to their similarity. The second class, termed feature-pivot methods, group together terms according to their co-occurrence patterns. The third class, probabilistic topic models, treats the problem of topic detection as a probabilistic inference problem.

In the context of feature-pivot methods, a straightforward way to consider simultaneous co-occurrence patterns between more than two terms is to apply FPM techniques [2]. FPM involves a set of techniques that were developed to discover frequent patterns in a large database of transactions.

Another technique is extracting candidate keywords and removing meaningless words by comparing candidate keywords in terms of ranking results [3]. Nevertheless, other research proposes a new graph-based framework that builds a Topical PageRank (TPR) on word graph to measure word importance with respect to different topics. After that, given the topic distribution of the document, they further calculate the ranking scores of words and extract the top ranked ones as keyphrases. Each of these methods has advantages and disadvantages. More techniques will be detailed in this report.

### 3 Overview of the Topic detection methods

We want to exploit each keyword extraction technique that can be used for tracking topics over time. In our work, keywords are a set of significant words in a post that gives high-level description of an event.

In this section, we give a brief description of some useful topic detection methods as well as the one we proposed.

#### 3.1 Latent Dirichlet Allocation:

LDA is probably the most popular probabilistic topic model; it uses hidden variables that represent the per-topic term distribution and the per-document topic distribution. Learning and inference in LDA is typically performed using Variational Bayes [4] but other approaches such as using Gibbs sampling have appeared [5].

Every document is considered as a bag of terms, which are the only observed variables in the model. The topic distribution per document and the term distribution per topic are instead hidden and have to be estimated through Bayesian inference.

#### 3.2 Document-pivot topic detection:

The method that we examine is an instance of a classical Topic Detection and Tracking method that uses a document-pivot approach.

In this approach, an incoming document or tweet is assigned to the same cluster as its most similar document, as long as their similarity is above some threshold. If it is not, a new cluster is created. In order to speed up the search for the most similar document, an implementation of Locality Sensitive Hashing LSH is made. The merit of using LSH is that it can rapidly provide the nearest neighbours with respect to cosine similarity in a large collection of documents.

#### 3.3 Graph-based feature-pivot topic detection:

This is an instance of a feature-pivot method, i.e. it groups together terms instead of documents. This approach in particular, organizes terms according to their co-occurrence patterns in a graph and applies a community detection algorithm (SCAN) on the graph, in order to come up with the resulting set of topics. The set of terms that will be used to construct the graph is selected



using the ratio of likelihood of appearance in the provided corpus over the likelihood of appearance in a reference corpus. In short, the algorithm steps are the Selection where the top K terms are selected and a node for each of them is created in the graph G, linking in which pairwise similarities between all pairs of terms are computed. The next step is the clustering and finally the connectivity of each of the hubs detected by SCAN to each of the communities is checked and if it exceeds some threshold, the hub is linked to the adjacent cluster(s). This step is called the cluster enrichment.

### 3.4 Soft Frequent Pattern Mining:

The soft frequent itemset mining method is another feature pivot method. It attempts to take into account co-occurrence patterns of degree larger than two (as compared to the graph-based approach). Like the graph based approach, soft frequent itemset mining selects the terms that will be processed. It consists of three different phases: Term selection which involves selecting a set of K terms from the corpus that will be grouped, Co-occurrence-vector formation and finally the post-processing that post-processes the results of the main part by removing duplicate topics. The clustering is based on distances between n-grams or clusters of n-grams. From the set of distances, those not exceeding a distance threshold are assumed to represent the same topic.

### 3.5 BNgram:

This is another feature-pivot approach. Its main goal is to find emerging topics in post streams by comparing the term frequencies from the current time slot with those of preceding time slots and by raising the importance of proper names by using a boost factor. It utilizes the df-idft measure to quantify the burstiness of n-grams and subsequently clusters the burstiest n-grams. In fact, it indexes all keywords from the posts of the collection. The keyword indices, implemented using Lucene, are organized into different time slots. In addition to single keywords, the index also considers bigrams and trigrams. Once the index is created, the df-idft score is computed for each n-gram of the current time slot based on its document frequency for this time slot and penalized by the logarithm of the average of its document frequencies in the previous t time slots.

### 3.6 Popularity:

We propose an novel keyword extraction technique. This model is composed of four steps mainly. The first step is extracting the top retweeted tweets as they should tell an interesting story. In the next step, we split the text of the tweets into a set of keywords then we deleted punctuation and stop words appearing in it. In the final step, we remove duplicates and we kept only two distinct topics as we considered that a set of keywords extracted from a tweet can represent a topic.

## 4 Topic detection framework

It is almost impossible to extract keywords manually due to the volume of the data generated. For a rapid extraction, we need to establish an automated process that detects keywords from the tweet files. We will use a software package that contains Java implementations of the topic detection methods that are presented in the previous section.

Since the data could be noisy, it will be the subject of some pre-processing steps such as the Tokenization which removes words that are common in a particular message but which don't have any useful meaning like stop words, punctuation, mentions..

### 4.1 Methodology

The methodology we followed is as follows: First, we produce a dataset of tweets for the same timeslots in which the ground truth dataset was extracted. Second, we use a Topic Detection Framework provided by the Social Sensor project [1] to extract topics from this dataset using different methods. Finally, we evaluate these methods based on some metrics by using the Topic Evaluator tool explained in the next chapter.

### 4.2 Preparing the Dataset

The first step was to prepare files containing a continuous Twitter stream for specific timeslots. We started with a large dataset collected from the public streaming API of Twitter which covers the English Football Association Challenge Cup Final. The FA Cup, is an annual knockout cup competition in English football; it is the oldest association football competition in the world. We chose this dataset because it has been shown that it has more relevant tweets and less entropy which make easier the topic detection task. The file contains a list of 444303 tweets. Essentially, the tweets were delivered for different periods. Most of them are talking about the final of the 2012 FA Cup in May 5 where the 2 finalists were Chelsea and Liverpool. Watched by a crowd of 89,102, Ramires put Chelsea in front in the 11th minute after he disposed Liverpool midfielder Jay Spearing and beat Pepe Reina in the Liverpool goal. They extended their lead in the 52nd minute when striker Didier Drogba scored. Liverpool substitute Andy Carroll scored in the 64th minute to reduce the deficit to one goal. Carroll thought he had scored a second in the 81st minute, but his header was saved on the line by Chelsea goalkeeper Petr Čech. Carroll ran off celebrating, as he thought the ball had crossed, but referee Phil Dowd did not award the goal and Chelsea held on to win the match 2–1.

Other tweets are dated from 2009 and 2011 and also a day before and a date after the event. The activity profiles of the trimmed intervals are depicted in Figure 1.

That's why we start this project by explaining the process of creating the database with the help of a Java program that we developed for this purpose.

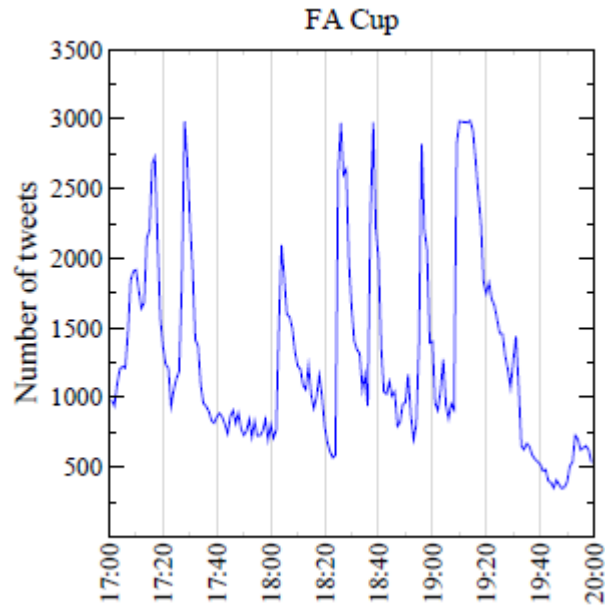


Figure 1 : Twitter activity during FACup event

#### 4.1.1 Dataset creation:

In order to gather files containing only tweets occurring in very short intervals, we opted for a Java project that, given a date and time, creates a Json file containing the tweets in the specified timeslot.

We generated 13 files corresponding to the timeslots of one minute in which the ground truth were created. The timeslots are:



Moreover, these files must have the same filenames as the corresponding files in the ground truth folder. We also computed another 13 files for the previous timeslots in order to be used for the BNgram method.

The big Json file is of 1.2 GB. However, the size of the files extracted from is between 2000 KB and 8000 KB and have about 600 to 3000 tweets.

#### 4.1.2 Difficulties faced:

First, we used some existing tools like File Query and many others available tools on the web. But, no one gave us the required result since they were not able to read correctly the structure of the Json files which make it impossible to build queries over it. That's why, we opted for a Java project which takes as an input the two dates that define a timeslot, extracts tweets from the big Json file which correspond to this timeslot and puts them in a new Json file.

Obviously we were not able to read the whole file into RAM and treat it because of its size. Fortunately, we found an API called Gson which is very efficient and flexible and solves the problem of parsing the big file.

In the Twitter response, dates have this format: "Sat Apr 04 06:15:55 +0000 2009" . So, we create a format String having the same format of the Twitter date "EEE MMM dd HH:mm:ss ZZZZ yyyy" and then we parse dates according to this particular pattern.

### 4.3 Implementation

In order to generate the result files, the parameters of the TMM jar file have to be changed in order to select which topic detection algorithm will be run, on which file and determine various output options. We fix the number of topics N to 2. And then, we simply run this command:

```
java -jar TMM.jar
```

### 4.4 Evaluation and results:

We obtain for each method 13 files for each timeslot. Each file has 2 lines which consist of a set of keywords that represent two different topics.

A sample of topics detected using each of the five methods can be seen in the following tables :

Time	Topics (N = 2)
16:16	drogba norton inside graham night didier sl kicked game commentator chelsea liverpool anthem national fans final jeering sad kick jeered
16:26	fans anthem national cup final drogba great reds game football chelsea ramires liverpool goal pick sl de gol ramirez del
16:41	chelsea liverpool kalou great run defence box ambushed mazy shoot downing liverpool fans ramires game anthem national suarez goal brilliant
16:53	mikel gerrard liverpool yellow card chelsea de ramires suarez booked liverpool chelsea game spearing amp henderson downing fans win cup

17:01	liverpool chelsea final henderson game spearing ramires el half team agger chelsea yellow mikel time liverpool card fans sl suarez
17:03	final half cup fa game shit de boring el la chelsea liverpool half time ht ramires win retweet wembley goal
17:18	chelsea win retweet suarez time kick tunnel luis game reds half liverpool chelsea sl kicked ramires final la de cup
17:24	chelsea goal drogba liverpool sl game finish good play great drogba wembley didier de drog scores fuck gol el la
17:25	drogba chelsea liverpool de la final didier el cup gol drogba chelsea goal sl wembley liverpool game didier scores finish
17:36	carroll andy liverpool goal game chelsea carroll back scores scored la de drogba en el liverpool didier chelsea final es
17:46	sl today de la en carroll attendance cech save el liverpool chelsea carroll game torres andy win suarez reds time
17:56	save cech great de decision gol carroll suarez andy la line goal liverpool carroll cech sl header technology chelsea super
18:09	chelsea cup fa win champions de la liverpool blue wembley chelsea liverpool final sl whistle congratulations great di congrats game

TABLE 1 : Results automatically detected by LDA

Time	Topics (N = 2)
------	----------------

16:16	sang jeered anthem lfc jeering fans national jackman89 henrywinter cfc sl cfcwembley facupfinal chelseafc kicked
16:26	cfcwembley facup facupfinal chelseafc ramires whatastart chelsea ktbfffh cfc sl cfcwembley facupfinal chelseafc goal chelsea
16:41	ambushed cfcwembley facupfin facup defence liverpool mazy chelseafc bni46 kalou messi fucking chelseaaanalysis liverpool making brilliant salomon cfc kalou
16:53	kick will_hoe sharing banter facupfinal terry luis suarez john tunnel 2006 adverso facup últimas chl 2010misterchip remontando finales liv ganó
17:01	sl cfcwembley mikel agger facupfinal stoppage chelseafc yellow fouling chelsea kick will_hoe sharing banter facupfinal terry luis suarez john tunnel
17:03	easportsfifa facup facupfinal retweet 2012 soccerdotcom win cfc chelsea chelseaaanalysis cfcwembley ciscmakassar facup chelseaindo facupfinal sunchelsea chelseafc ciscbandung ktbfffh
17:18	2nd sl cfcwembley facup facupfinal chelseavliverpool half chelseafc kicked easportsfifa facupfinal retweet chelseafc win slygarba tommyyusuf chelsea
17:24	sl cfcwembley facupfinal drogba chelseafc goal chelsea coyb cfcwembley facup toogood facupfinal drogba kramtanner yeahbitch chelseafc cfc
17:25	didier sl cfcwembley chelseafc_sa facupfinal oooh drogba chelseafc goal chelsea africanbeast cfcwembley facup wholetthedrogout facupfinal gemmaaalou drogba chelseafc sberthiaumeespn cfc



17:36	didier sl cfcwembley chelseafc_sa facupfinal oooh drogba chelseafc goal chelsea africanbeast cfcwembley facup wholetthetrogout facupfinal gemmaaalou drogba chelseafc sberthiaumeespn cfc
17:46	didier sl cfcwembley chelseafc_sa facupfinal oooh drogba chelseafc goal chelsea africanbeast cfcwembley facup wholetthetrogout facupfinal gemmaaalou drogba chelseafc sberthiaumeespn cfc
17:56	sl cfcwembley cech facupfinal liverpool saved carroll chelseafc claiming header lifeofsamuel leanneemcdonald cech facup facupfinal jaspers08 tweeting_keith sambora71 mogeorgeeastend shanedenfreude
18:09	sl cfcwembley cech facupfinal liverpool saved carroll chelseafc claiming header lifeofsamuel leanneemcdonald cech facup facupfinal jaspers08 tweeting_keith sambora71 mogeorgeeastend shanedenfreude

TABLE 2 : Results automatically detected by Doc-p

Time	Topics (N = 2)
16:16	sl #facupfinal @chelseafc #cfcwembley @chelseafc #cfcwembley kicked kicked drogba graham inside norton drogba inside night graham norton
16:26	sl #facupfinal @chelseafc chelsea goal #cfcwembley @chelseafc #cfcwembley goal chelsea ramires chelsea ramires
16:41	messi @chelseanalysis liverpool messi salomon kalou fucking fucking @chelseanalysis making liverpool kalou making brilliant salomon #cfc ambushed liverpool mazy @chelseafc great run kalou #facup ambushed @chelseafc great mazy defence shoot box liverpool #cfcwembley box #cfcwembley shoot run defence

16:53	mikel yellow card card obi mikel mikel
17:01	phil player dowd's daniel foul phil notebook notebook agger mikel dowd's @thefadotcom #facupfinal chelsea @chelseafc 1 yellow fouling mikel chelsea @chelseafc time #facupfinal stoppage min sl agger yellow time 1 agger mikel stoppage fouling #cfcwembley min 1-0
17:03	#lfc #facup
17:18	sl 2nd #facupfinal 2nd half half @chelseafc kicked @chelseafc #cfcwembley kicked #cfcwembley #cfc
17:24	sl #facupfinal @chelseafc chelsea goal #cfcwembley @chelseafc #cfcwembley goal chelsea #facupfinal drogba 2-0 chelsea liverpool drogba 2-0 chelsea
17:25	#facup el en de sl #facupfinal @chelseafc chelsea goal #cfcwembley @chelseafc #cfcwembley goal chelsea
17:36	#facupfinal chelsea 2-1 liverpool 2-1 #lfc carroll andy andy
17:46	sl 89,102 #facupfinal @chelseafc 89,102 attendance attendance today's #cfcwembley @chelseafc #cfcwembley today's #facupfinal forces save suarez @chelseafc forces cech cech @chelseafc suarez #cfcwembley save
17:56	sl #facupfinal claiming liverpool line saved line carroll @chelseafc header claiming header super cech @chelseafc cech liverpool carroll #cfcwembley #cfcwembley super saved technology ruined line goal line technology game goal football times joke joke @tommybowe14 @tommybowe14 ruined times

18:09	sl #facupfinal final chelsea 2 liverpool final 1 @chelseafc whistle chelsea 2 @chelseafc 1 liverpool #cfcwembley whistle #cfcwembley cup beat wembley cup competition @premierleague chelsea liverpool beat #lfc fa win wembley fourth years 2-1 chelsea fa fourth liverpool @premierleague years #cfc competition
-------	---

TABLE 3 : Results automatically detected by BNgram

Time	Topics (N = 2)
16:16	sang henrywinter jeering anthem fans jeered national sang henrywinter jeering anthem fans jeered national
16:26	sang henrywinter jeering anthem fans jeered national sang henrywinter jeering anthem fans jeered national
16:41	kalou liverpool defence great mazy ambushed box shoot run kalou liverpool defence great mazy ambushed box shoot run
16:53	adverso últimas 2006 liverpool resultado liv ganó remontando chl finales adverso últimas 2006 liverpool resultado liv remontando ganó chl finales
17:01	stoppage yellow fouling card mikel agger stoppage yellow fouling chelsea card mikel agger
17:03	time courtesy goal wembley half chelsea mins ramires final lead liverpool strike cup time half wembley goal chelsea ramires final lead liverpool cup
17:18	underway 2nd kicked half underway kicked 2nd half
17:24	drogba liverpool goal chelsea drogba liverpool goal 52nd chelsea

<b>17:25</b>	chelsea 52nd drogba liverpool goal chelsea 52nd drogba liverpool goal
<b>17:36</b>	marca 2007 historia historico record primer drogba jugador cuatro finales didier marca 2007 historia historico record primer jugador drogba cuatro finales didier
<b>17:46</b>	goles sólo equipo desventaja ganó carroll liv años chl últimos remontando goles sólo equipo desventaja ganó carroll liv años chl últimos remontando
<b>17:56</b>	liverpool header cech super carroll line claiming saved liverpool cech header super carroll chelseafc line claiming saved
<b>18:09</b>	comeback whistle chelsea final liverpool congratulations winning comeback whistle chelsea final liverpool winning

TABLE 4 : Results automatically detected by SFIM

We observe that the BNgram considers the Hashtags and mentions as keywords and SOFT\_FIM got many keywords repeated in the two lines.

## 5 Topic evaluator

This [section shows how](#) we compute scores to test the performance of the topic detection algorithms described above.

To evaluate each method, we compared their output with ground truth using 3 metrics which are as follows:

- Topic recall :Percentage of ground truth topics that were successfully detected by a method. A topic was considered successfully detected in case the automatically produced set of keywords contained all mandatory keywords for it and none of the forbidden.

$$topic\_recall = \frac{\text{number of mandatory keywords detected}}{\text{number of ground truth topics}}$$

- Keyword precision : Percentage of correctly detected keywords out of the total number of keywords for the topics that have been matched to some ground-truth topic. The total precision of a method is computed by micro-averaging the individual precision scores over all matched topics.

$$term\_precision = \frac{\text{number of correctly keywords detected}}{\text{number of candidate topic terms}}$$

- Keyword recall :Percentage of correctly detected keywords over the total number of keywords of the ground truth topics that have been matched to some candidate topic. The total recall is similarly computed by micro-averaging.

$$term\_recall = \frac{\text{number of correctly keywords detected}}{\text{number of actual topic terms}}$$

### 5.1 Running the Evaluator

Although this tool turns out to be simple to run, it needs some prerequisites and it suffers from a number of limitations. First, the results files for all target timeslots must all be in the same directory and must have the same filenames as the corresponding files in the ground truth directory. Otherwise the script will produce an error message and stop. Second, the format of the ground truth files was not clear and not explained. In fact, each line is a ground truth topic. Each topic has three sets of words: required, optional and forbidden. These three sets of words are separated with a tab character from each other. In order to get a match, a topic must have all

required words but none of the forbidden. Optional words only count for keyword recall and precision and not for topic recall and precision. Finally, the words in each set of words are separated with a semicolon and when there are alternatives for some word, these are placed in square brackets and separated with a space.

To run this evaluation script, we need just to execute this command

```
java -jar TopicEvaluator.jar GroundTruthsDirectory ResultsDirectory StartN EndN StepN
```

where StartN is the first @N value for which precision / recall will be evaluated, EndN is the last @N value for which precision / recall will be evaluated and StepN is the step to increase the @N value.

## 5.2 Evaluation and Results

The basic idea behind the popularity method is to consider that the most retweeted tweets in a timeslot are the one that contain the keywords that represent a trending event. By looking at the distribution of tweets according to how much they were retweeted in 05/05/2012 at 16:16 displayed in Figure 2, we can see that most of the tweets were not retweeted. However, one has been retweeted more than 1600 times. Then, this tweet should certainly give relevant information.

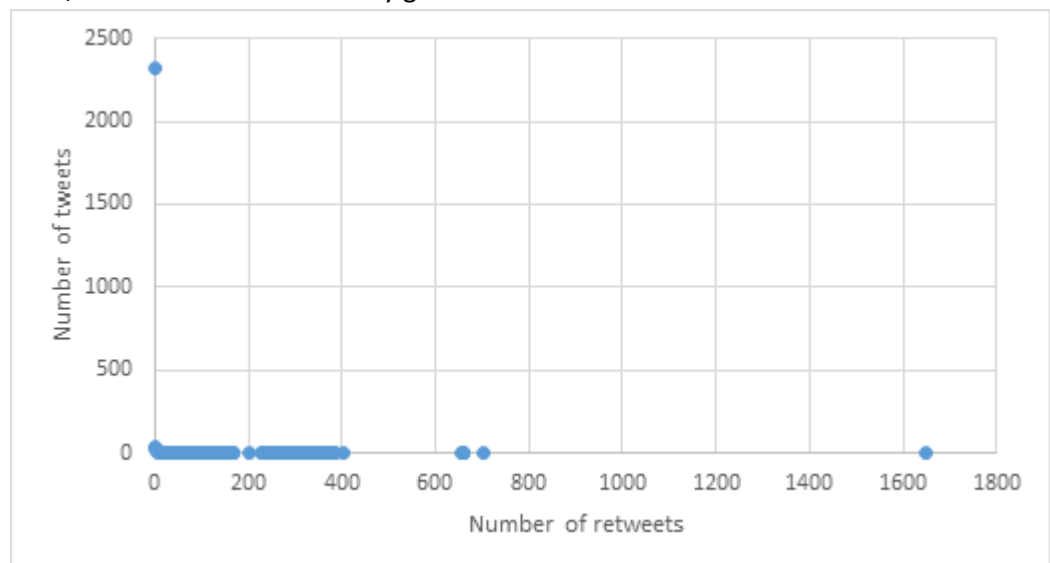


Figure 2 : curve of the distribution of the tweets (according to their popularity)

The code of this project can be found in this Github repository :

<https://github.com/chaimabr/TwitterAnalyser>

The topics returned by the popularity method for the same timeslots as previously cited are given in Table 5.

Time	Topics (N = 2)
16:16	subs carroll maxi kuyt carragher doni shelvey kelly leesiemaszko plse reds fans facupfinal chelseascum disrespect ynwa httpcoxqfwhkk
16:26	chelseafc team liverpool cech cole lamps mata drog cfcwembley facupfinal henrywinter fans sang national anthem many didnt even jeering they jeered
16:41	chelseafc team liverpool cech cole lamps mata drog cfcwembley facupfinal henrywinter fans sang national anthem many didnt even jeering they jeered
16:53	misterchip liverpool ltimas finales facup remontando resultado adverso west chelseafc chelsea goal cfcwembley facupfinal
17:01	henrywinter fans sang national anthem many didnt even jeering they jeered leesiemaszko plse reds fans facupfinal chelseascum disrespect ynwa httpcoxqfwhkk
17:03	chelseafc chelsea goal cfcwembley facupfinal chelseafc ramires right foot shot puts front mins thanks mata assist cfcwembley facupfinal
17:18	chelseafc half time here wembley courtesy ramires strike after mins cfcwembley facupfinal easportsfifa retweet think chelsea facupfinal
17:24	easportsfifa retweet think chelsea facupfinal chelseafc even scoreboard cant keep ramires cfcwembley facupfinal httpcocrizft

<b>17:25</b>	easportsfifa retweet think chelsea facupfinal henrywinter fans sang national anthem many didnt even jeering they jeered
<b>17:36</b>	easportsfifa retweet think chelsea facupfinal henrywinter fans sang national anthem many didnt even jeering they jeered
<b>17:46</b>	easportsfifa retweet think chelsea facupfinal willhoe john terry luis suarez sharing some banter tunnel before kick facupfinal httpcowghxwv
<b>17:56</b>	mariobalotelad taken pitch replaced andy carroll know youre wrong facupfinal willhoe john terry luis suarez sharing some banter tunnel before kick facupfinal httpcowghxwv
<b>18:09</b>	easportsfifa retweet think chelsea facupfinal willhoe john terry luis suarez sharing some banter tunnel before kick facupfinal httpcowghxwv

TABLE 5 : Results automatically detected by Popularity method

If we choose one of these timeslot and compare the keywords obtained to the ground Truth, we will find a significant similarity between both. For example, the ground Truth for the second timeslot 16:26 have the following text :

[ramires chelsea];[goal 1-0 1 0] score;yes

As explained before the first set of words are required. We can see them in the topics extracted. For the second set of words, they are optional. One of them is extracted which is the word 'goal'. The third set of words are forbidden and they don't appear in this case. We can conclude at this step that the popularity method is respecting the rules of the ground Truth.

The performance of the 6 methods as presented in the following table:

<b>Method</b>	<b>topic recall</b>	<b>term precision</b>	<b>term recall</b>
<b>LDA</b>	0.308	0.3	0.6
<b>DOC_PIVOT</b>	0.538	0.309	0.531
<b>GRAPH_BASED</b>	0	0	0



<b>SOFT_FIM</b>	0.538	0.392	0.606
<b>BNgram</b>	0.846	0.337	0.569
<b>Popularity</b>	0	0	0

TABLE 6 : Comparison of Topic Detection Algorithms

According to this table, we can conclude that BNgram method has the best topic recall while SOFT\_FIM achieved better term precision and term recall. This indicates that SOFT\_FIM is able to retrieve more target topics and that it also represents them in a quite complete and accurate manner. The Graph-based and POPULARITY display both null metrics. They did not detect any correct topics and keywords.

		LDA	Doc-p	SFIM
<b>N = 1</b>	Topic recall	0.308	0.462	0.308
	Term precision	0.225	0.3	0.2
	Term recall	0.429	0.643	0.533
<b>N = 2</b>	Topic recall	0.923	0.923	0.846
	Term precision	0.275	0.267	0.264
	Term recall	0.6	0.582	0.58

TABLE 7 : Comparison of the metrics for different values of @N

The different metrics at a range of values for @N where N is the number of topics, is displayed in Table 7. For both values, Doc-p method achieves higher metrics values. Interestingly, SFIM produces a smaller number of topics than the other approaches, so the metrics value are lower. However, topic recall is very high for all the methods for higher values of N. It is also important to notice that mainly all the values have increased by increasing N.

## 6 Topic Viewer

This chapter discusses the design and implementation of a collector of tweets and a topics extraction system based on java platform. By using the TMM, we perform the topics extraction part. Besides, the collection of tweets is based on a collector system published by the Social Sensor project.

The system functions are implemented with Java language in Eclipse environment. The system is featured with expandability, reliability, suitability and friendly interface and plays positive practical value in tweets collection and keywords extraction.

A Github repository to this project can be found in this link :

<https://github.com/chaimabr/TopicViewerWeb>

### 6.1 Features supported and input data :

The design requirements and the features supported of Keyword extraction and tweets collection is as follows:

- simple interface, easy operation, stable running;
- system functions include entering keywords separated by ‘,’ and the duration of the extraction of tweets process. It also enables choosing the name of the file in which the result will be saved and indicating the name of the creator of this new data. This is all for the tweets collector. For the second part, the system gives the possibility to choose one of the existent files, the duration of the timeslot and the method to be applied to the data.
- the topics should appear in a timeline centred in the first date where the tweets appear.



The screenshot shows a web form for collecting tweets. It has four input fields: 'KeyWords' with the text 'Football, Tunisia, CharlieHebdo ...', 'Duration' with a dropdown menu showing '1 min', 'File name' with the placeholder 'Enter a file name ...', and 'User name' with the placeholder 'Enter your name ...'. Below these fields is a blue button labeled 'Collect tweets'.

Figure 3 : First part of the Topic Viewer (The collector)

Choose a file	
Timeslot size	
Method	LDA
Launch Extractor	

Figure 4 : Second part of the Topic Viewer (The extractor)

## 6.2 System Structure Design

The primary functions based on the system requirements analysis are tweets collection, topics extraction and topic visualisation.

- Tweets collection module: the module requires information from the user to be sent to the twitter API that will return the tweets. The system records created file name, creation time and the name of the user.
- Topic extraction module: every extraction of keyword has to be implemented under task management. Users can select the method that will be applied. The module is also featured with historical task query.
- Topic visualisation module: it refers to a timeline which display the keywords extracted for each topic ordered from the first date where the tweets were collected to the last date.

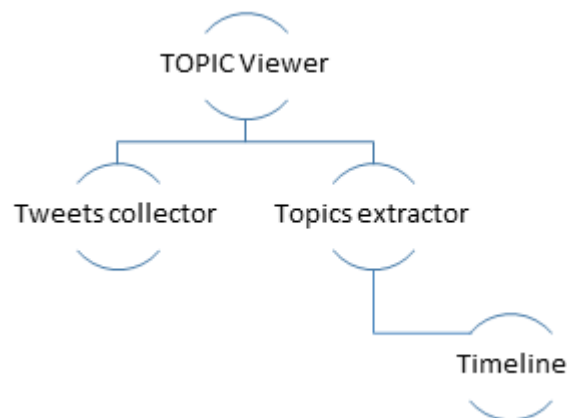


Figure 5 : The structure of the Topic Viewer

## 6.3 Software Architecture

### *Tweet collection Module*

To run the java code, we used Apache Tomcat [link?](#) which is an open source web server and servlet container. It implements the Java Servlet and the JavaServer Pages (JSP) specifications and provides a pure Java HTTP web server environment for java code to run.

This module needs the interaction of the user to define for how much time the collection will be performed and for what event. The definition of the event will be done by specifying a certain number of keywords. Next is to wait for the twitter collector to return tweets.

The process is shown in Figure 1.



Figure 1 : Application background process for collecting tweets

- 1- The user makes a request for tweets
- 2- Server issues request to Twitter dataset collector
- 3- Twitter collector returns response and results are received
- 4- The result is returned to the user

### *Topic Extraction Module*

This module is an implementation of the TMM. In this library, for each of the 5 algorithms, there is a java package that contains, among other classes, a class named TopicDetector and this class has a public method named createTopics. For each of these topic detection methods, there is a parameter file, with a fixed name, which has to reside on the execution folder. To adjust the parameters of each method, these files are the only way to do it independently from the java project. All these classes will run under Apache Tomcat server. In the following, it will be presented how this module is working.

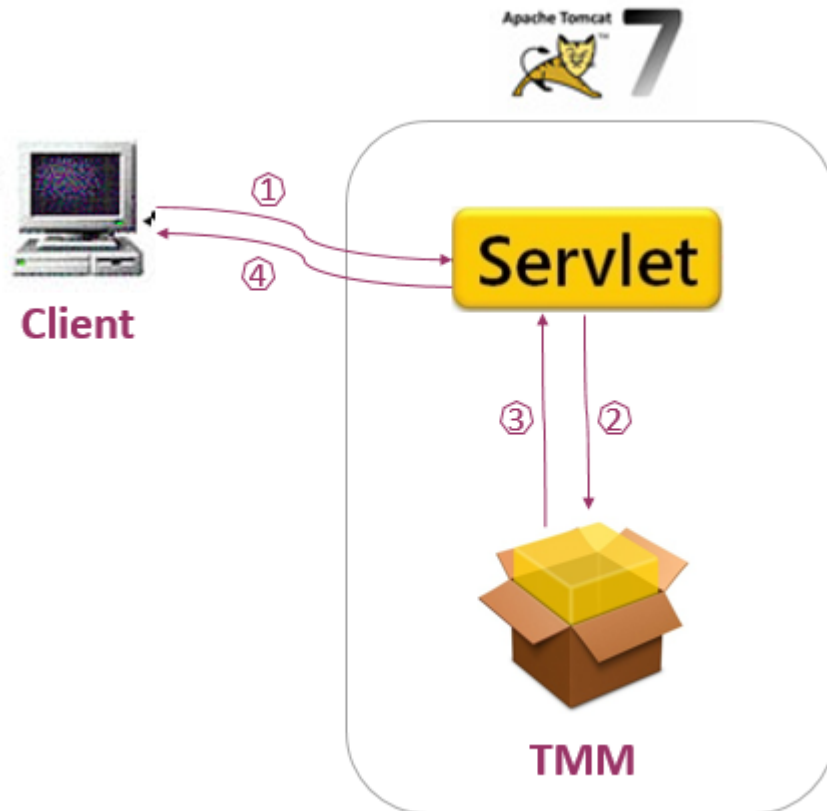


Figure 4 : Application background process for extracting topics

- 1- The user chooses a file, the duration of the timeslot and a method
- 2- Server runs the TMM
- 3- The TMM applies the desired method on the data provided and returns the result
- 4- The Topics are displayed to the user in a timeline

That was the big picture. In figure 5, a deep version of each component of the Apache tomcat box is shown to understand that component easier.

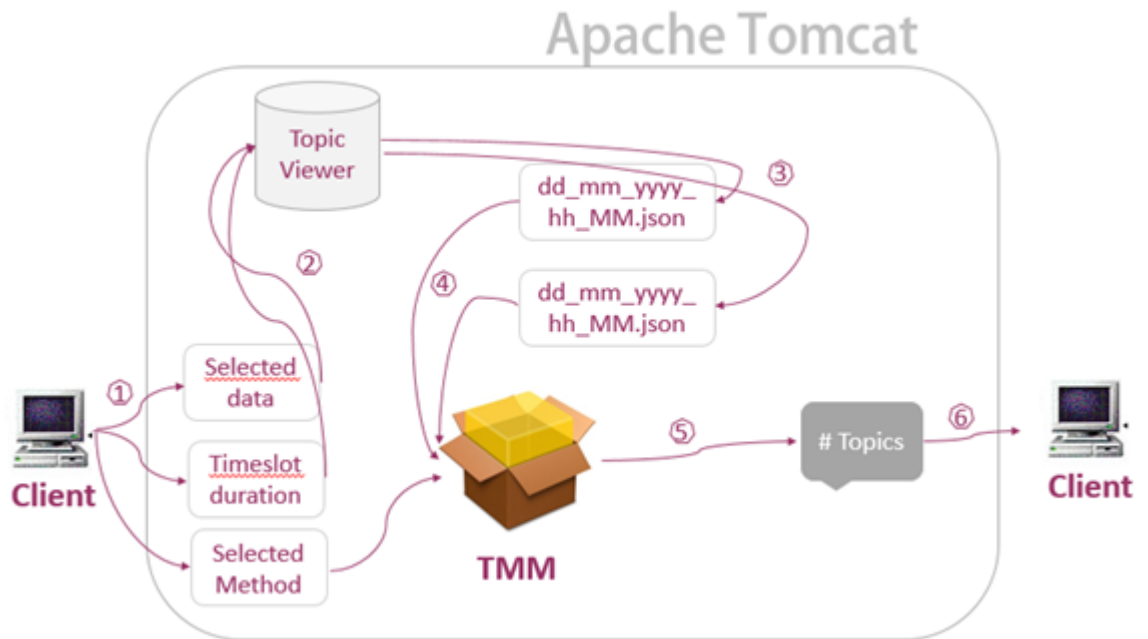


Figure 5 : Internal working of the apache Tomcat box

- 1- The client send request containing the tweets file, the timeslot size and the method
- 2- The tweets file and the timeslot size are sent to the Topic Viewer and the method is sent to the TMM
- 3- For each timeslot, two files are retrieved: one for the required timeslot and another for the previous one and are named based on their date and time
- 4- The 2 files are sent to the TMM
- 5- The TMM apply the method selected on the file(s) and return the topics as a result
- 6- The topics are displayed to the user in a timeline

### *Topic Visualisation Module*

This module is related to the previous one. It enables the visualisation of the topics returned by the Topics extractor. And because topics occur at a point in time or during a segment of time, we used a timeline, a friendly way to temporal data visualizations. We implement SIMILE timeline, an open-source web widgets. This timeline contains one or more Bands, which can be panned infinitely by dragging with the mouse pointer, using the mouse scroll-wheel or the keyboard's arrow buttons. A band can be configured to synchronize with

another band such that panning one band also scrolls the other. For our case, we chose 5 bands: Year, Month, Day, Hour and Minute.

The topics will appear in the Minute band preceded by an icon while the other bands are configured to be just overview bands which contain small event markers.

Here is the resulting timeline for topics related to an Apple conference event obtained by applying the LDA with one minute timeslots :



Figure 6 : SIMILE timeline

## 6.4 Topic Viewer implementation

### 6.4.1 Tweets Collection Module

Before developing this module, the first idea was to use a twitter API. We studied the advantages and drawbacks of the search and streaming API and decided to use the search API. Despite the fact that the Streaming API usually returns a much higher flow of tweets, about 1% of the full firehose of tweets, the Search API can collect a wider range of data and has more powerful queries. However, the tests have shown that the Search API gives a relevant number of tweets but in only one range of time 01:57 to 01:59 whatever the time of the extraction and an index of recent 6-9 days of tweets.

From this, we decided to use the twitter dataset collector provided by the SocialSensor project which make use of the Twitter Streaming API. It facilitates the distribution of Twitter datasets by downloading sets of tweets (if still available) using their ids and a number of keywords as input. It is significantly simpler to use since it has very few dependencies (jsoup and twitter4j) and simple text-based input/output file formats.

### 6.4.2 Keywords Extraction Module

As said previously, this module is an implementation of the TMM.jar library. The user will select the file on which the TMM will be performed. This file needs to be split into two smaller files corresponding to a timeslot and the previous one for each iteration. The duration of these new file is simply the one that the user will select as an input.

#### 6.4.3 Topic Visualisation module

The topics are returned from the TMM and then exported to an xml file in a format readable by the timeline widget. The default date time parser uses the Javascript Date parser built into the browser. An Event Source controls the loading of data sources into a timeline. The xml format used for the timeline includes the above attributes:

- wiki-url
- wiki-section
- date-time-format

Here an example of the xml file :

```
<data
  wiki-url="http://simile.mit.edu/shelf/"
  wiki-section="Simile JFK Timeline">
  <event
    start="Sat May 20 2015 00:00:00 GMT-0600"
    title=" Here the text of the topics">
  </event>
  <event>
    ...
  </event>
</data>
```

#### 6.5 System evaluation and conclusion

To evaluate the Topic Viewer, we have looked for a trending event and collected tweets in real time. The choice was based on what is most emerging in Twitter. Since the football is one of the hottest topics, we picked the South American Football Championship known as Copa America when Brazil was playing against Peru. An injury-time goal from Douglas Costa gave Brazil a barely deserved 2-1 win over a plucky Peru side on Sunday. The goal of Neymar in the fifth minute had been the subject of many tweets during and after the match. Figure 7 shows that the date we picked was appropriate to get relevant tweets.



## Tweets per day: copaAmerica

May 16th — June 15th

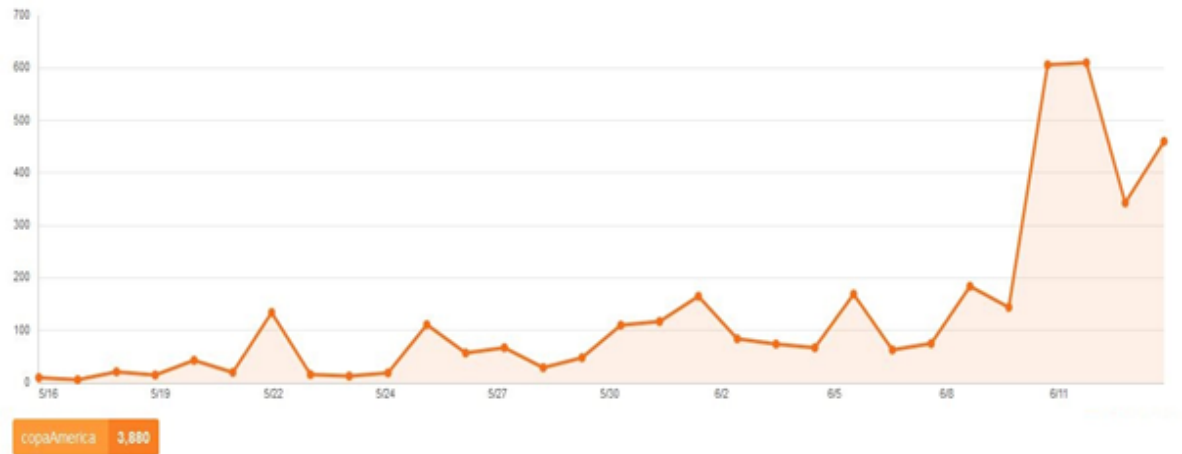


Figure 7 : Distribution of tweets per day for the Hashtag 'copaAmerica'

The Copa America is an international football competition contested between the men's national football teams of CONMEBOL, determining the continental champion of South America. It is the oldest international continental football competition. This year, it was held in Chile.

The analysis of this event on Twitter 20 minutes before we started the collection are given in figure 8.

Twitter Collection Started: 6/15/2015 12:20 am



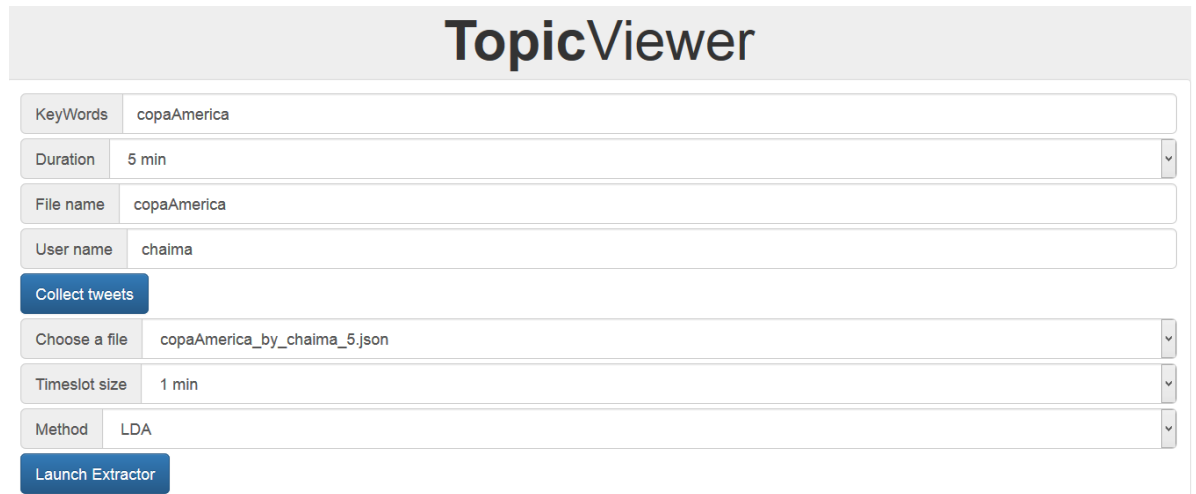
Figure 8 : Twitter analysis of 'copaAmerica' 20 minutes before the collection

We collected tweets for 5 minutes using 'copaAmerica' as a keyword which correspond to the most used Hashtag that time and we got a Json file of 27427 KB containing 6056 tweets.

In the Topic detection phase, we chose this file and run the collector on it for the different methods with a timeslot equal to one minutes. The file was

split into 5 files, each one corresponding to a timeslot and of size between 3MB and 4MB of about 1300 tweets approximately. It is clear that the distribution of the tweets on these files is practically uniform.

The interface looks like the following during the extraction :



The screenshot displays the 'TopicViewer' web interface. It features a header with the title 'TopicViewer'. Below the header, there are several input fields and buttons. The fields are labeled 'KeyWords', 'Duration', 'File name', 'User name', 'Choose a file', 'Timeslot size', and 'Method'. The values entered in these fields are 'copaAmerica', '5 min', 'copaAmerica', 'chaima', 'copaAmerica\_by\_chaima\_5.json', '1 min', and 'LDA' respectively. There are two buttons: 'Collect tweets' and 'Launch Extractor'.

Field	Value
KeyWords	copaAmerica
Duration	5 min
File name	copaAmerica
User name	chaima
Choose a file	copaAmerica_by_chaima_5.json
Timeslot size	1 min
Method	LDA

Buttons: Collect tweets, Launch Extractor

Figure 9 : Topic Viewer interface

## 7 Discussion and Conclusion

In this first part of this report, we described the whole process from building a repository of needed Json files to the modality of detecting topics with different methods and we finished by testing the performance of these methods which proved their effectiveness in detecting newsworthy topics but still need further work to fine-tune it. Future improvements can include other features besides n-grams co-occurrences and the  $df-idf_t$ .

Nevertheless, a novel approach is proposed to improve keyword extraction. The proposed method rely on the fact that tweets that have been more retweeted are more eligible to represent an emerging topic. The method does not prove efficiency in terms of topic and keyword recall and keyword precision since it is based on untested assumptions. However, it could be combined with other methods to shape up the task of extraction. For example, we can apply it first to get the top 100 retweeted tweets and then apply one of the other method only on these tweets.

In the second part, we talked about a web java project that we called Topic Viewer which implements the collection of tweets and the extraction of topics from it in the same interface. The topics are displayed in a timeline after being extracted. This interface still need future work to improve its design and to add more methods to be used for the extraction. Since the code of the TMM was not provided and the documentation lacks many details, it was not possible to ameliorate the quality of the extraction and make all the methods work without so much investigation and time consuming.

Finally, this project offered me valuable learning opportunities and helped me to work more on my development skills. I wish to express special thanks to my supervisors Mr. Raphaël Troncy, Mr José Luis Redondo Garcia and Mr Giuseppe Rizzo for their generous support throughout the whole project, for their patient and meticulous supervision, and for their precious advice.

## References

- [1] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, A. Jaimes. Sensing trending topics in Twitter. *IEEE Transactions on Multimedia* (pre-print), 2013
- [2] B. Goethals. Frequent set mining. In *The Data Mining and Knowledge Discovery Handbook*, chapter 17, pages 377–397. Springer, 2005.

- [3] Sungjick Lee and Han-joon Kim. News Keyword Extraction for Topic Tracking. In Fourth International Conference on Networked Computing and Advanced Information Management, 2008 , Page(s): 554 - 559
- [4] C. W. Fox and S. J. Roberts. A tutorial on variational bayesian inference. Artificial Intelligence Review, 38(2):85-95, 2011.
- [5] H. Xiao and T. Stibor. Efficient collapsed Gibbs sampling for latent dirichlet allocation. In Asian Conference on Machine Learning (ACML), volume 13 [2] of JMLR W&CP, Japan, 2010.