# Towards a Twitter Observatory: A multi-paradigm framework for collecting, storing and analysing tweets

Ian Basaille, Sergey Kirgizov, Éric Leclercq, Marinette Savonnet, Nadine Cullot

Laboratoire LE2I - UMR6306 - CNRS - ENSAM
Univ. Bourgogne Franche-Comté
9, Avenue Alain Savary
F-21078 Dijon - France
Email: Firstname.LastName@u-bourgogne.fr

***Abstract*—In this article we show how a multi-paradigm framework can fulfil the requirements of tweets analysis and reduce the waiting time for researchers that use computational resources and storage systems to support large-scale data analysis. The originality of our approach is to combine concerns about data harvesting, data storage, data analysis and data visualisation into a framework that supports inductive reasoning in multidisciplinary scientific research. Our main contribution is a polyglot storage system and a set of tools that can provide a suitable solution for mixing different types of algorithms in order to maximise the extraction of knowledge. We describe the software architecture of our framework and we show how it has been used in major projects and what characteristics have been validated.**

***Index Terms*—polyglot storage, massive datasets, knowledge discovery, open source software, Twitter analysis**

## I. Introduction and motivations

In general, analysing massive datasets requires different types of algorithms with different theoretical foundations such as graph theory, linear algebra, or statistical models. Regarding tweets analysis, intrinsic links, such as hashtags, user mentions, and retweets, have a strong impact on the storage model and on the performance of the algorithms being used. Nevertheless, addressing a scientific question often requires mixing different classes of algorithms using different data models that retrieve data from different storage structures. For instance, graph-based algorithms using a matrix adjacency representation are useful for finding communities, Laplacian matrix is useful to evaluate centrality [1]. In general, graph-based algorithms are near-sighted: taking into account contextual information is costly. Linear algebra algorithms can be used to identify clusters using Singular Value Decomposition (SVD) or Principal Component Analysis (PCA). They are also used to identify complex relationships or interaction patterns among entities in multidimensional spaces. Machine learning algorithms and statistical models are used to predict links or behaviours, to detect anomalies or events. In a context of massive datasets, the most efficient storage structure should be used according to the selected algorithms, but different kinds of algorithms are usually mandatory.

In the context of tweets, analysis can be performed at different levels of granularity: at an individual level like gender detection, sentiment analysis, e.g. extraction of features that are not explicit; and at a corpus level, e.g. emergence of groups of individuals having a similar behaviour. Thus, possible outcomes of the analysis are the discovery of social structures, social positions, i.e. role of individuals.

Applications that supports tweets analysis are directed towards either building new knowledge or decision-making. They can be classified into three types according to their objectives: 1) influencing or controlling the real world (decision oriented applications) such as customer relationship management or response to emergency situations in the case of natural disasters or pandemics; 2) detecting changes in the real world such as predicting an earthquake; and 3) building new knowledge, in a scientific context, without direct impact in/from the real world. In the next paragraphs we give some real examples of such types of applications.

Recently, researchers have started studying the use of Twitter during natural crises or in emergency situations. These studies show that information shared during crises can have a potential value for crisis managers, who can use this information to improve their operational response to the crisis. Mendoza et al. in [2] have investigated the behaviour of Twitter users during a major earthquake in Chile in 2010. The TEDAS (Twitter-based Event Detection and Analysis System) [3] platform aims to: 1) detect new events (crime, car accident, etc.); 2) analyse the spatial and temporal aspects of an event; 3) identify the importance of an event.

Some researchers try to identify, by analysing tweets, events that occur in the real world; for instance, in [4], authors have tried to estimate the location of an earthquake epicentre. In [5] the authors concentrate on changes in the frequency of tweets to observe the status of the real world during the Great Eastern Japan Earthquake on March 2011. In [6] the authors show how to predict dark triad personality traits from Twitter usage and a linguistic analysis of tweets, thus they can detect some high-risk behaviours.

In knowledge construction area, Burnap and al. [7] have built models that predict the information flow size and survival (retweets) on Twitter following the terrorist event in Woolwhich, London in 2013. In [8], authors show how to use Twitter to mine public health information. They focused on producing data that correlate with public health metrics and knowledge. The target of knowledge extraction can be Twitter itself. In [9] authors have analysed the ways in which hashtags spread on a network defined by the interactions among Twitter users.

As a conclusion, this short review shows that applications which perform tweets analysis require different classes of algorithms to discover different properties in data sets. Moreover, properties or structures found in data sets are scale and time dependent (such as community detection) thus in order to extract them, an iterative and incremental approach is fundamental. Furthermore, to convert information into usable knowledge, discovered structures must be characterised by key features which describe or explain structures.

Our contribution is an open source framework SNFreezer[1] that supports the management and the analysis of social data with different paradigms. We developed a set of specific components: 1) a polyglot storage to store and retrieve tweets in different structures that are able to scale up with the data flow requirements; 2) a set of tools that can be combined to analyse data and extract knowledge.

The remainder of the paper is organised as follows. While section 2 discusses works related to multi-paradigm storage, section 3 describes the software architecture of the framework, and compares our proposal with related works. Section 4 relates experiments and results obtained in different projects. We will describe how two multi-disciplinary projects have been used to test the scalability and robustness of the polyglot storage system. We also show how our tools can perform an iterative analysis starting from an events detection followed by community detection and their characterisations using hashtags.

## II. State of the art

Most enterprise business applications rely on relational database management systems (RDBMS). This technology is mature, widely understood and successfully deployed. However, some concerns have recently became apparent: 1) RDBMS may not have adequate performance for massive datasets; 2) RDBMS cannot provide the scalability required by high-throughput web applications; 3) the structure of the relational model can be too rigid or not relevant to deal with the variability of complex data networks such as online social networks. Explicit and implicit links between social data can be a hindrance to the use of RDBMS if they are combined with massive data.

Considering these drawbacks, a number of new systems, not following the relational model paradigms, have recently emerged. They are often denoted under the umbrella term of NoSQL databases [10]. Their common features are scalability and flexibility in the structure of data. NoSQL database management systems provide different solutions for specific problems: the volume of datasets is addressed in the column-oriented NoSQL or key-value (HBase, Cassandra); documents and links management is supported by document databases (CouchDB, MongoDB); high density of links, nodes and properties are taken into account in graph database management systems (GDBMS) which are also ideal for performing queries that walk down hierarchical relationships (Neo4j, HypergraphDB). XML oriented databases provide a highly extensible data model but lack scalability in the context of social networks.

NoSQL databases are accessed by different APIs, so Atzeni and al. [11] propose a common programming interface to NoSQL systems supporting application development by hiding the specification details of the various systems. The TinkerPop project[2] adopts a similar approach for graph databases. It introduces a graph query language, Gremlin, which is a domain-specific language based on Groovy[3], supported by most GDBMS. Unlike most query languages, Gremlin is an imperative language focusing on graph traversals.

The multi-paradigm principle tends to generalize these different approaches, exploring a different way than the abstraction. In modelling, multi-paradigm approaches address the necessity of using multiple modelling paradigms to design complex systems [12]. Indeed, complex systems require the use of multiple modelling languages to: 1) cope with the inherent heterogeneity of such systems; 2) offer different points of view on all their relevant aspects; 3) cover different activities of the design cycle; 4) allow reasoning at different levels of detail during the design process [13]. As a result, multi-paradigm modelling addresses three orthogonal directions of research: 1) multi-formalism modelling, concerned with the coupling and transformation between models described in different formalisms; 2) model abstraction concerned with the relationship between models at different levels of abstraction; 3) meta-modelling concerned with the description of classes of models dedicated to particular domains or applications called *Domain Specific Languages* (DSL).

Multi-paradigm data storage or polyglot persistence uses multiple data storage technologies, chosen according to the way data is used by applications and/or algorithms [14]. As Ghosh states in [15], storing data the way it is used in an application simplifies programming and makes it easier to decentralize data processing. ExSchema [16] is a tool that enables automatic discovery of data schema from a system that relies on multiple document, graph, relational, column-family data stores.

Our objective is to define a multi-paradigm framework that

[1] https://github.com/SNFreezer/

[2] http://www.tinkerpop.com/

[3] http://groovy.codehaus.org/

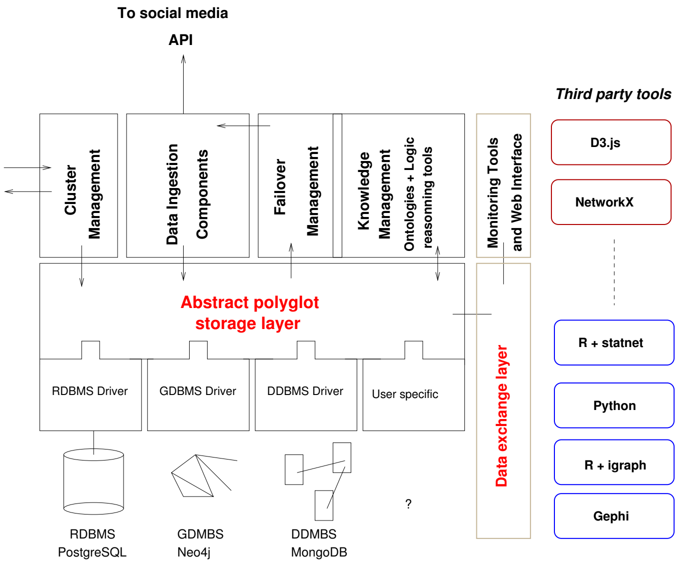


Figure 1. Software architecture

aims at reducing development and analysis time by choosing storage systems (eventually multiple) that are the most relevant to the problem/algorithm; and a set of tools to allow users to set up analysis to produce knowledge from data in an iterative and incremental process. We must perform different analyses with different algorithms, so we break the problem into segments and apply different database models.

## III. An overview of our software architecture

We started by analysing existing solutions and we retained the project YourTwapperKeeper[4] (YTK), an open source project, that claims to provide users with a tool that archives data from Twitter directly on a server. After a period of tests and code review, we identified some major drawbacks. This tool was not able to collect tweets in various languages; the choice of the database engine limits the volume of datasets; it does not retrieve information on accounts such as the list of following/followers nor the timeline of the users. Thus, we chose to enhance YTK with a real storage layer and to add connectors with analysis tools.

### A. *Polyglot Storage*

To address the problem of tweets storage, both in terms of performance and interoperability (easy connection of third party tools), as well as the adequacy between algorithms and data structures, we have specified and developed a storage layer using YTK. The polyglot persistence storage layer includes relational databases (RDBMS), a graph data store (GDBMS), and a scalable storage system (DDBMS) that can be used simultaneously (figure 1).

Compared to polyglot database approaches [17], our approach allows to duplicate information in different storage systems according to the planned analysis. Hook points and interfaces defined in the abstract storage layer allow developers to add their specific drivers for other storage systems. The data gathering is provided by the data ingestion module connected to the *Streaming* and *Search APIs* of Twitter using two main processes. Two other processes are used to collect the followers/following information at a defined frequency and, at the launch of a harvest, to retrieve all the possible tweets from the timeline of the users. According to his needs and type of information to be analysed, one can choose between various storage structures:

- In the case of structured information, a relational schema can be used (figure 2). The *Tweet* table contains tweets and retweeted tweets. *Tweet* is connected by foreign keys with hashtags (*Tweet_Hashtag*), URLs (*Tweets_URL*), symbols (*Tweet_Symbol*) and query sources (presence of a keyword, a hashtag or an account in the tweet) (*Tweet_Source*). *Retweet* and *Mention* are tables that represent relationships between users and tweets. *Retweet* table contains the user that retweeted, the ID of the retweeted tweet and the date. *Mention* table connects a tweet with the mentioned users. *User* and *Identity* tables represent information about users. The relational database is implemented in PostgreSQL.

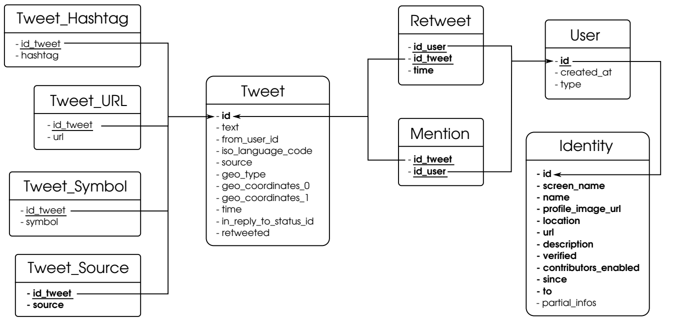


Figure 2. An extract of a simplified relational logical data model for tweets without followers and timelines

- In the case of the study of linked information (for example social network type data structure), a graph database is suitable (figure 3). Objects (tweets, users, hashtags, etc.) are nodes with properties, and relationships are described by edges with properties. We implemented this schema in Neo4j.
- In the case of high traffic, it is preferable to store information in a non-normalized database scheme (one table for tweets and a few others for followers) or in JSON files or in MongoDB.

The choice of storage backends can be cumulative (both relational normalized and JSON files for example). We also propose a set of tools that implement model transformation to asynchronously transform data from one storage system to another. A specific data exchange layer is dedicated to application services, and third party tools are plugged in according to the expected analysis. Connectors have been developed to analyse data with third party software such

[4] https://github.com/540co/yourTwapperKeeper

as *R* and *igraph*[5] and to display results with *D3.js*[6]. If a connector is not available, the layer provides export files for the different classes of algorithms such as files containing adjacency matrix, graph triple, multidimensional array or CSV (Comma-Separated Values) for spreadsheet.

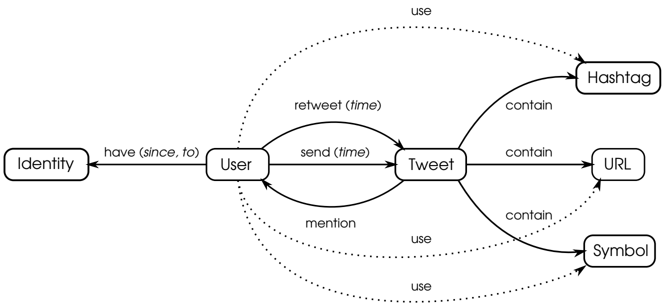


Figure 3. An extract of a simplified graph data schema for tweets

### B. Cluster and failover mode

Cluster and failover modes have been developed to overcome some limitations of the *Streaming* and *Search APIs* of Twitter. For instance, one application can only retrieve tweets matching one of the conditions, i.e., query source, in a set of 400. Each account on Twitter can define applications, but for each IP address hosting the application the previous limitations apply. However, some projects require more than 400 query sources. To fulfil this requirement we have developed a cluster management module.

In the cluster mode, several virtual machines, in which instances of the data harvesting processes are deployed, can be used in a project to collect tweets using matching criteria in a set of several thousands of query sources. Query sources can be defined on each virtual machine using a web interface or can be imported automatically using scripts and import files submitted to one of the virtual machine that acts as a master. In the cluster mode, the assumption of high volume of tweets is taken, thus the storage layer can use a non normalized schema on PostgreSQL, or can store data in MongoDB or in JSON files.

In the case of interruption during the harvest, the process in charge of the *Search API* can retrieve tweets for a period of seven days. Automatic emails are sent to the administrator with a notification of the failure (storage, networks etc.) and when restarting the harvesting process, a comparison with the latest tweet ID is done and the process in charge of the *Search API* starts collecting lost tweets. Moreover, it is possible to use a local database on each virtual machine using MySQL and a global database using PostgreSQL.

To monitor the harvest and the instances of the cluster, some tools were also developed. They allow, amongst other things, to get the number of tweets per period of time, the most used hashtags or the most active users and to send alerts by email to researchers or administrators. In the case of high volume of tweets, a replica database is used to calculate indicators for the monitoring tools. The figure 4 gives an example of the deployed cluster to collect tweets during European Election campaign of 2014.

### C. Comparison of our framework with existing ones

In this section we compare the different features of our framework with some concurrent projects. Table I presents the different projects and criteria retained.

Sciences Po Medialab[7] project provides individual components to retrieve, store, analyse and display social graphs. It is a set of tools largely independent that are hard to interconnect without IT programming skills. The tool used to collect tweets is basic with no advanced feature such as cluster or failover.

In [18], authors describe an integrated service for analysing social media on demand. Their platform COSMOS proposes the use of a suite of open source tools for text and network analysis. They focus on on-demand and longitudinal analysis of public opinion and sentiment, around a socially significant event. They use other sources than Twitter, but do not support failover or cluster mode to collect data.

In [3], authors describe a system specialised for events detection. Their platform TEDAS provides a visualisation interface, but their crawler is very simple. As their main focus is on event detection, other types of analysis are not available yet.

These platforms have specific goals but don't cover a broad spectrum like our framework does. Our framework lacks an integrated user interface, as our main goal is to provide a multi-paradigm framework with multiple pluggable tools that can be used in various ways (separately, together, etc.) and not an integrated platform. We think that user interaction is domain dependant and we leave the choice of user interface and combination of tools to the users of our framework in order to build what we call an *observatory*.

To the best of our knowledge, there is no approach that blends a Twitter crawler tool with advanced features such as cluster mode to overcome various limitations of Twitter API's, a failover mechanism to make sure no tweet is missed during the harvest; with the possibility of plugging different analysis modules that can be sequenced to perform incremental and iterative analysis.

## IV. Experiments and results

The experiments performed with our framework have two main goals: 1) to validate the polyglot storage (extensibility, cluster mode, failover); 2) to validate the multi-paradigm analysis approach that produces knowledge in an iterative and incremental way.

The first project is TEE 2014 (A Comparative International Study of the Use of Twitter by Candidates at the European Parliamentary Elections in May 2014) that aims at studying political tweets, across six countries, during the European Parliament Elections in 2014.

[5] http://igraph.org/

[6] http://d3js.org/

[7] http://tools.medialab.sciences-po.fr

Table I
COMPARISON OF OUR FRAMEWORK WITH SOME CONCURRENT PROJECTS

| | Our framework | Medialab | COSMOS | TEDAS |
|---|---|---|---|---|
| Integrated User Interface | No | No | ✓ | ✓ |
| Polyglot storage | ✓ | No | No (Language detection) | No (Location prediction) |
| Back-end storage independence | ✓ | No | No (MongoDB, distributing data using Hadoop) | No |
| Cluster Mode | ✓ | No | 1 connection to Twitter, analysis in parallel | No |
| Multiple Twitter API | ✓ | ✓ | No (Streaming API and other sources than Twitter) | No |
| Failover mechanism | ✓ | No | No | No |
| Sequencing analysis | ✓ | No | 9 types of analysis available (tweet or corpus level), but no sequencing | Only event detection and analysis |
| Multiple type of pluggable analysis | ✓ | ✓ | No | Only event detection and analysis |

The second project is the study of events, users communities and users interactions, for Social Customer Relationship Management (CRM) applications, in collaboration with a private company. Its goal is to provide a set of tools for Community Managers and Data Analysts to help them to analyse marketing campaigns as well as the behaviour of customers and prospects on online social networks.

### A. *Validation of polyglot storage, cluster and failover modes*

The validation of storage in relational database (normalised and non-normalised schema), graph database, cluster and failover took place in the TEE2014 Project. This project focuses on tweets from the accounts of all candidates, in Belgium, France, Germany, Italy, Spain and the United Kingdom, along with the messages addressed publicly to them. Moreover, major hashtag-related conversations associated with the elections in each country were followed.

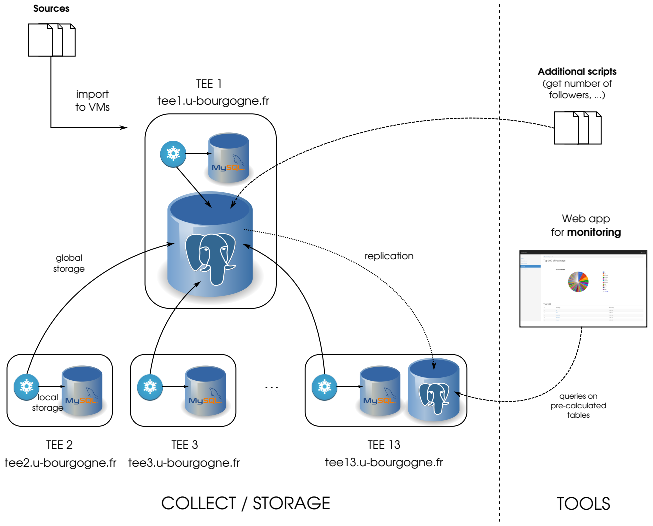


Figure 4. Architecture used in cluster mode for TEE 2014 project

Thirteen virtual machines have been used to support the important number of query sources ($\simeq$ 5000) and the *Streaming API* limitations that do not allow more than 400 query sources per instance (figure 4). More than 50 million of tweets have been collected during a month. The size of the database using a non-normalised relational schema is around 50GB and the size of the normalised relational schema with indexes is approximatively the same. The storage layer has proved to work well in a cluster mode. We also verified dynamic addition of query sources and error recovery through the failover component. Multilingual tweets including Cyrillic, for studying how political tweets in EU and in Ukraine are linked, were also gathered. According to the IT team of one of the partners, the waiting time between the collection and the start of analysis has been significantly reduced (from several weeks to less than a week) from the use of YTK.

The NoSQL storage in MongoDB and in JSON files has been tested by collecting tweets during the *2014 FIFA World Cup*: an average of 1 million tweets per day were collected, with peaks at the end of the event and for particular games. During the whole event, more than 1.1 billion of tweets were stored both in JSON files and in MongoDB. The collection size was 3.2TB. Twice, the flow of tweets exceeded the threshold of 1% of total traffic and the failover mechanism was triggered successfully to collect the missing tweets.

### B. *Evaluation of the multi-paradigm analysis approach*

In this subsection, we discuss for each project the results obtained using a multi-paradigm analysis. The corpus of TEE 2014 has been analysed by different research teams in the different countries involved, in order to explore three general questions:

- How political parties use Twitter as a communication tool?
- Which specific hashtags are used during the campaign, is it possible to identify distinct periods or events?
- Which relations exist between identified users (candidates) and other users?

In the context of CRM, tweets about carpooling have been gathered. We analysed about 8M tweets and tried to find and identify or characterise communities, events and studied how each carpooling actor is using tweets on a specific event.

The large size of Twitter network and its dynamics make it very difficult to directly work with it. In order to defeat the complexity of the data, we propose an iterative approach (realised in the form of a web-application).

This web application allows scientists: 1) to perform interactive explorations on the dataset by detecting interesting time intervals (possible real-world events) and 2) to characterise these intervals. To achieve the first objective, we propose a method based on kernel density estimation [19], [20]. For the second objective, we use several powerful algorithms and simple statistics in order to characterise the detected time intervals. The number of such algorithms is not limited, and they can be added as plug-ins to our system.

After describing the event detection algorithm, we briefly discuss two of the characterisation methods used: 1) community detection in the hashtag-user network; 2) hubs and authorities in the graph of retweets.

*1) Exploratory event detection:* Many existing event detection algorithms use statistical analysis of time-series [21], [22], [23], or NLP-based techniques [24], [25]. But before applying any statistical algorithm we have to make some hypotheses about the prior distribution of our data. In many cases the justification of such choice is quite difficult. We propose to use a simple method that finds local minima (maxima) of temporal density of tweets. This method finds the moments after which the number of published tweets begins to increase (or decrease). The method has only one parameter (bandwidth) that allows to smoothly change the mode of the event detection: when the bandwidth is small micro-events can be detected, and when it is large only macro-events appear.

This method, by itself, does not give any description of the event, except its start and end times. Thus, in order to describe an event, we split the data using hashtag-decomposition: for each hashtag of interest, we draw temporal density plot, and we combine all these plots in the same image, that allows us to compare them. The difference in hashtag frequencies can be regarded as a first approximation of the semantic event description. A part of program source code (detection of local minima and maxima of temporal density) is available on github[8]. For data retrieval and analysis tasks we used *R*. We also used *R* as a web-server (packages *httpuv, mime, webutils, xtable, RPostgreSQL*). The visualisation part has been created with the help of *R* package *streamgraph* [9], that produces *D3.js*-code.

A screen-shot of our web application is shown at figure 5. The x-axis represents time, and y-axis shows the frequency of tweets that contain hashtags of interest. Peaks correspond to the periods of use of most frequent hashtags and different colours represent different hashtags. Two ellipses correspond to the two main events: 1) election day and 2) result announcement day. Potential events (grey vertical lines) are the timestamps after which there is a rapid increase (or decrease) in the number of tweets. Using two vertical green bars users can select a time interval of interest, and perform a detailed analysis of tweets (by pressing the button "MAKE ME HAPPY!") from the selected period; such as calculating top 10 hashtags, top 10 users, showing community structure, describing hubs and authorities, etc.

A lot of useful information can be extracted from Twitter datasets using this web application. Here, we only consider the peak between the two thick vertical bars presented at figure 5. We can see that #TellEurope and #NowSchulz hashtags have become very popular during this period, and beyond the period people almost never used these hashtags. This was due to TellEurope debate organised by European Broadcasting Union at this moment. Martin Schulz participated in the debate.

*2) Hashtag-user network: community detection:* In this second experiment, we tried to identify communities from a subset of official political accounts. To do so, we used the Louvain modularity based algorithm with the most used hashtags (filtered by a binomial test to remove non significant hashtags). An ontology of political domain knowledge has been defined, it includes political parties, their region, and official accounts. We compared communities members found by the algorithm with known affiliation described in the ontology. Some singularities have been detected. To help social scientists analyse singularities, we enhanced the visualisation by assigning different colors to members and communities. The result is depicted in figure 6, hashtags are represented in yellow circles, and users' color depends on their political party. We can see that, most of the time, users of the same party tend to use the same set of hashtags. However, the figure puts in evidence a singularity: in the right lower corner, some users (orange nodes) in a political party are using the same hashtags than users of another party (red nodes).

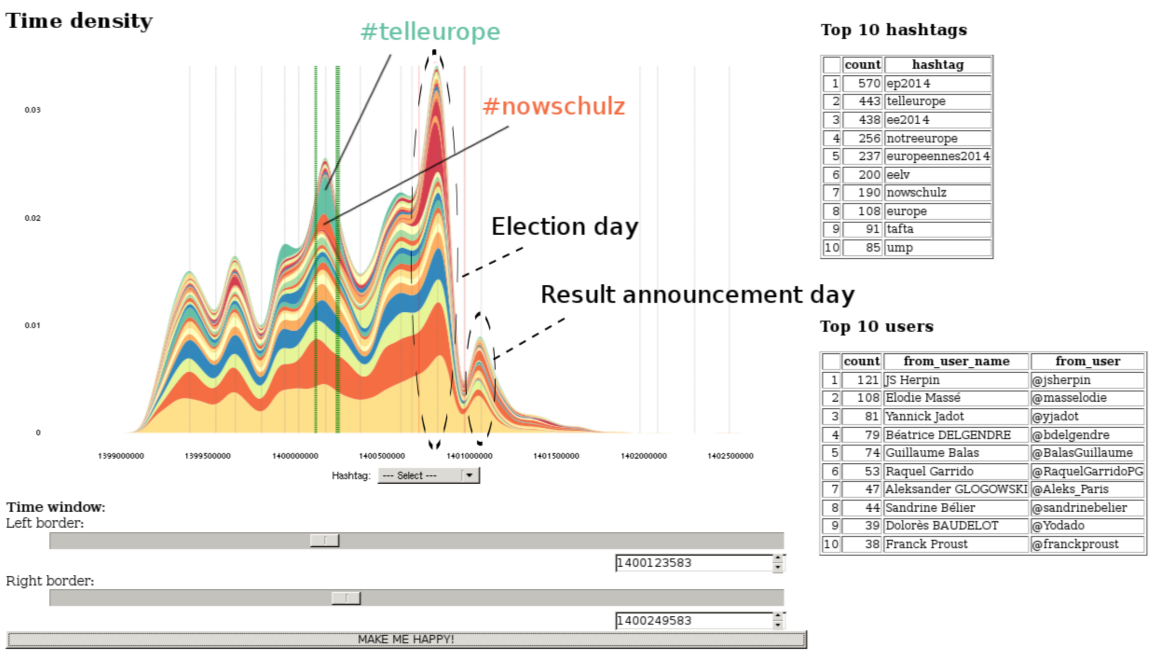
*3) Graph of retweets: hubs and authorities:* This experiment was directed towards the discovery of hubs and authorities in the graph of retweets. This analysis is based on the well-known HITS algorithm [26] that is used to compute the hubs and authorities in web pages. Given an adjacency matrix $A$, hub and authority scores are given respectively by the principal eigenvector of $AA^T$ and $A^T A$. The principal eigenvectors can be computed by a Singular Value Decomposition (SVD). In our case, $A$ is a square matrix of users built with retweets.

We have filtered users that have retweeted more than 20 times tweets from identified politicians. The scored hubs and authorities produced by the algorithm are then analysed by the social scientists. It shows some specificities that have not been found using traditional database analysis (e.g., frequency of hashtags, number of retweets per users). Using traditional analysis, the social scientists noticed all small parties behave in a similar way. However, the top 10 ranking of hubs and authorities is predominantly monopolised by members and politicians of one small party. In order to investigate a specific user behaviour, a component of the web application can be used (figure 7).

For the CRM project, we studied a particular event, iden-

[8] Removed for double-blind review.

[9] https://github.com/hrbrmstr/streamgraph



**Top 10 hashtags**

| | count | hashtag |
|---|---|---|
| 1 | 570 | ep2014 |
| 2 | 443 | telleurope |
| 3 | 438 | ee2014 |
| 4 | 256 | notreeurope |
| 5 | 237 | europeennes2014 |
| 6 | 200 | eelv |
| 7 | 190 | nowschulz |
| 8 | 108 | europe |
| 9 | 91 | tafta |
| 10 | 85 | ump |

**Top 10 users**

| | count | from_user_name | from_user |
|---|---|---|---|
| 1 | 121 | JS Herpin | @jsherpin |
| 2 | 108 | Elodie Massé | @masselodie |
| 3 | 81 | Yannick Jadot | @yjadot |
| 4 | 79 | Béatrice DELGENDRE | @bdelgendre |
| 5 | 74 | Guillaume Balas | @BalasGuillaume |
| 6 | 53 | Raquel Garrido | @RaquelGarridoPG |
| 7 | 47 | Aleksander GLOGOWSKI | @Aleks_Paris |
| 8 | 44 | Sandrine Bélier | @sandrinebelier |
| 9 | 39 | Dolorès BAUDELOT | @Yodado |
| 10 | 38 | Franck Proust | @franckproust |

Figure 5. A part of twitter observatory interface

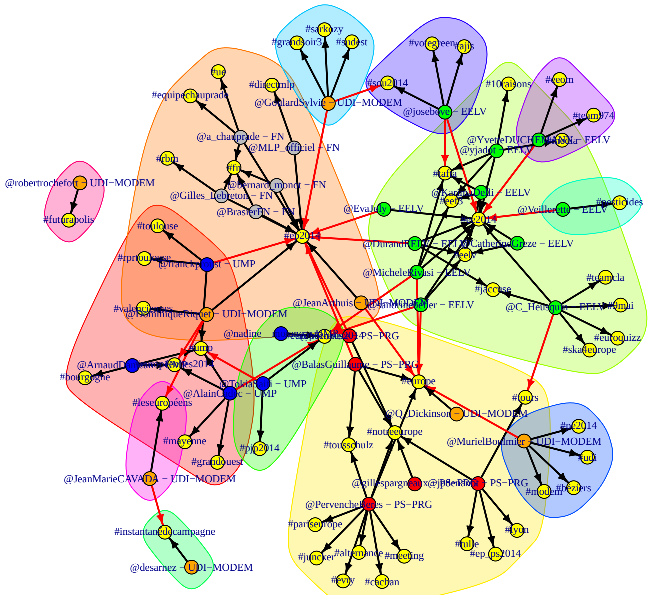


Figure 6. Communities and singularities found on the french corpus - TEE 2014

tified by using the method developed in [23]. This event was related to the taxi drivers blocking some roads around Paris, protesting against carpooling service *UberPop*. We performed two computations of hubs and authorities, the first one without filtering (table II) and the second one allowing only identified accounts to be retweeted. Those accounts were officials accounts of *Uber* and some of their french competitors that also operate in Paris (table III).

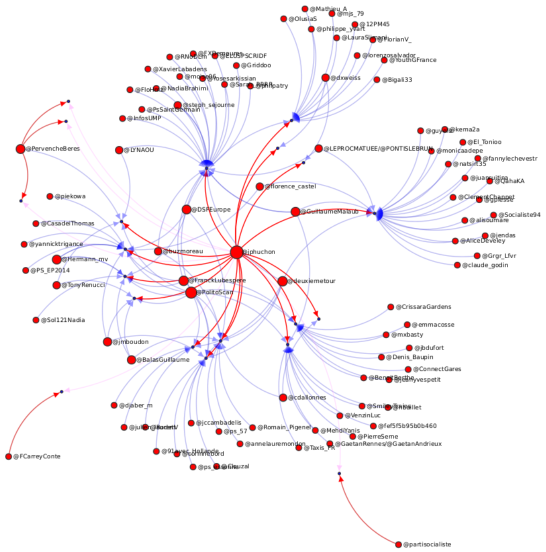


Figure 7. Tweets and retweets for a specific user

The first experiment shows accounts related to taxi drivers and associations, and people related to this profession as top authorities. Hubs are also related to the taxi community, along with several robot accounts. *Uber* and competitors other than taxis do not appear in these top 10s.

The second experiment shows some, but not every, followed accounts as hubs and authorities. The top hub is a robot account, and only one competitor of *Uber* appears as an authority. The top 5 is meaningful, but after that, there is not enough data to have meaningful results. That is, the followed

Table II
TOP 10 HUBS AND AUTHORITIES WITH EVERY ACCOUNTS AVAILABLE

| Authorities | Hubs |
|---|---|
| Taxi_de_Paris | 999Hha |
| ThibaudDELETRAZ | CGT_TAXIS |
| PierrePeyrard | StepouneTest |
| MonPereCeTaxi | taxiazur |
| del_tass | 94ALLADIN |
| InnovMobi | MoMoElTaCo |
| zaherinho | augenoux |
| ASSOCIATIONVTC | nabilakoff |
| nabilakoff | zaherinho |
| pumbijourdain | Le_Terrier |

accounts have not been retweeted enough. We can conclude that they have stayed *under the radar* during this event, and didn't take part in the discussions regarding the event.

Table III
TOP 10 HUBS AND AUTHORITIES WITH A RESTRICTION ON RETWEETED ACCOUNTS

| Authorities | Hubs |
|---|---|
| UberLyon | ConcoursRetweet |
| iDVROOM | UberLyon |
| Uber_Lille | Uber_Paris |
| Uber_Cannes | Uber_Lille |
| Uber_Paris | Uber_Cannes |
| Seatecawen | Seatecawen |
| patron_pme | patron_pme |
| nicolaslr | nicolaslr |
| Mbcustode | Mbcustode |
| MagalieBarreira | MagalieBarreira |

## V. CONCLUSION

In this article, we have presented an open framework integrating: 1) a polyglot scalable storage system to respond to the adequacy between analysis algorithms and storage systems; 2) a set of tools that includes event detection that helps scientists to conduct research on social interaction using graph analysis and to use tools with limited human effort. Several projects in an international context have proved the efficiency and robustness of the framework. The different analysis tools implemented in the framework allow to perform iterative and incremental analysis through a dedicated web application.

Our future work is directed towards adding new functionalities to the framework: 1) adding temporal representation of graphs to study the dynamics of communities, hubs and authorities; 2) to obtain a collaborative framework that supports inductive research methods and thus to store results from various analysis in the DBMS with traceability.

## REFERENCES

[1] X. Qi, E. Fuller, Q. Wu, Y. Wu, and C.-Q. Zhang, "Laplacian centrality: A new centrality measure for weighted networks," *Information Sciences*, vol. 194, pp. 240–253, 2012.

[2] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: Can we trust what we RT?" in *Proc. of the first workshop on social media analytics*. ACM, 2010, pp. 71–79.

[3] R. Li, K. H. Lei, R. Khadiwala, and K. K.-C. Chang, "TEDAS: A Twitter-based event detection and analysis system," in *Proc. of International Conference on Data Engineering*, 2012.

[4] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proc. of the 19th international conference on World wide web*. ACM, 2010, pp. 851–860.

[5] T. Sakaki, F. Toriumi, and Y. Matsuo, "Tweet trend analysis in an emergency situation," in *Proceedings of the Special Workshop on Internet and Disasters*. ACM, 2011, pp. 3–8.

[6] C. Sumner, A. Byers, R. Boochever, and G. J. Park, "Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets," in *Int. Conf. on Machine Learning and Applications (ICMLA), 2012*, vol. 2. IEEE, 2012, pp. 386–393.

[7] P. Burnap, M. L. Williams, L. Sloan, O. Rana, W. Housley, A. Edwards, V. A. Knight, R. Procter, and A. Voss, "Tweeting the terror: modelling the social media reaction to the woolwich terrorist attack," *Social Netw. Analys. Mining*, vol. 4, no. 1, 2014.

[8] M. J. Paul and M. Dredze, "You are what you tweet: Analyzing twitter for public health," in *ICWSM*, 2011, pp. 265–272.

[9] D. M. Romero, B. Meeder, and J. Kleinberg, "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 695–704.

[10] A. Moniruzzaman and S. A. Hossain, "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison," *arXiv preprint arXiv:1307.0191*, 2013.

[11] P. Atzeni, F. Bugiotti, and L. Rossi, "Uniform access to NoSQL systems," *Information Systems*, vol. 43, pp. 117–133, 2014.

[12] B. S. Hodge, S. Huang, J. D. Siirola, J. F. Pekny, and G. V. Reklaitis, "A multi-paradigm modeling framework for energy systems simulation and analysis," *Computers & Chemical Engineering*, vol. 35, no. 9, pp. 1725–1737, 2011.

[13] C. Hardebolle and F. Boulanger, "Exploring multi-paradigm modeling techniques," *Simulation*, vol. 85, no. 11-12, pp. 688–708, Nov. 2009.

[14] J. Sharp, D. McMurtry, A. Oakley, M. Subramanian, and H. Zhang, *Data Access for Highly-Scalable Solutions: Using SQL, NoSQL, and Polyglot Persistence*. Microsoft patterns & practices, 2013.

[15] D. Ghosh, "Multiparadigm Data Storage for Enterprise Applications," *Software, IEEE*, vol. 27, no. 5, pp. 57–60, September 2010.

[16] J. Castrejon, G. Vargas-Solar, C. Collet, and R. Lozano, "ExSchema: Discovering and Maintaining Schemas from Polyglot Persistence Applications," in *Software Maintenance (ICSM), 2013 29th IEEE International Conference on*, Sept 2013, pp. 496–499.

[17] P. J. Sadalage and M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison Wesley, 2012.

[18] P. Burnap, O. Rana, M. Williams, W. Housley, A. Edwards, J. Morgan, L. Sloan, and J. Conejero, "COSMOS: Towards an integrated and scalable service for analysing social media on demand," *Int. Journal of Parallel, Emergent and Distributed Systems*, 2014.

[19] M. Rosenblatt *et al.*, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[20] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, pp. 1065–1076, 1962.

[21] N. A. James, A. Kejariwal, and D. S. Matteson, "Leveraging cloud data to mitigate user experience from "breaking bad"," *arXiv preprint arXiv:1411.7955*, 2014.

[22] J. Bai and P. Perron, "Computation and analysis of multiple structural change models," *Journal of applied econometrics*, vol. 18, no. 1, pp. 1–22, 2003.

[23] R. Killick, P. Fearnhead, and I. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[24] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, 2013.

[25] A. Guille and C. Favre, "Mention-anomaly-based event detection and tracking in twitter," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM Int. Conf. on*. IEEE, 2014, pp. 375–382.

[26] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.