Elsevier Editorial System(tm) for Expert Systems With Applications
Manuscript Draft

Manuscript Number:

Title: MISNIS: An Intelligent Platform for Twitter Topic Mining

Article Type: Full Length Article

Keywords: Twitter;
Intelligent Topic Mining;
Fuzzy Fingerprints;
MongoDB;
MISNIS

Corresponding Author: Prof. Joao Paulo Carvalho, Ph.D

Corresponding Author's Institution: INESC-ID / Instituto Superior Tecnico, Universidade de Lisboa

First Author: Hugo H Rosa, MsC

Order of Authors: Hugo H Rosa, MsC; Gaspar Brogueira, MsC; Fernando Batista, Ph.D; Joao Paulo Carvalho, Ph.D

Abstract: Twitter has become a major tool for spreading news, for dissemination of positions and ideas, and for the commenting and analysis of current world events. However, with more than 500 million tweets flowing per day, it is necessary to find efficient ways of collecting, storing, managing, mining and visualizing all this information. This is especially relevant if one considers that Twitter has no ways of indexing tweet contents, and that the only available categorization "mechanism" is the #hashtag, which is totally dependent of a user's will to use it. This paper presents an intelligent platform and framework, named MISNIS, that facilitates these issues and allows a non-technical user to easily mine a given topic from a very large tweet's corpus and obtain relevant contents and indicators such as user influence or sentiment analysis.

Cover Letter

Lisboa, November 12th, 2015

Dear Editor-in-Chief

Please find attached the paper entitled “*MISNIS: An Intelligent Platform for Twitter Topic Mining*”, that we are submitting for possible publication in Expert Systems With Applications.

This paper presents an intelligent platform and framework that facilitates the collection, storage, management, visualization and intelligent mining of extensive Twitter corpora, and provides non-technical users an expert system to easily mine a given topic of interest and obtain relevant contents and indicators, such as user influence or sentiment analysis. As such, we believe that the work is of interest and within the scope of this journal.

Looking forward to receiving your news,

Sincerely,

João Paulo Carvalho
Hugo Rosa
Gaspar Brogueira
Fernando Batista

Mailing address:
INESC-ID
Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
E-mail: joao.carvalho@inesc-id.pt
Tel.: +351962148272

MISNIS: An Intelligent Platform for Twitter Topic Mining

Highlights:

- We develop and present an intelligent framework/platform to efficiently collect, store, manage and visualize large Twitter corpus.
- We develop a REST API interface that includes endpoints to visualize information and to allow other applications to easily interact with the database.
- We present and describe a platform add-on that implements an expert system for Twitter topic mining that can be used by social experts without computer technical knowledge. The system allows the detection of topic relevant tweets from a large collection of tweets and returns relevant contents and indicators such as user influence within the topic or sentiment analysis.
- We present a small case study as an application example.

MISNIS: An Intelligent Platform for Twitter Topic Mining

Hugo Rosa
hugo.rosa@inesc-id.pt
Gaspar Brogueira
gmrba@iscte.pt
Fernando Batista
Fernando.Batista@inesc-id.pt
Joao P. Carvalho
joao.carvalho@inesc-id.pt

Abstract

Twitter has become a major tool for spreading news, for dissemination of positions and ideas, and for the commenting and analysis of current world events. However, with more than 500 million tweets flowing per day, it is necessary to find efficient ways of collecting, storing, managing, mining and visualizing all this information. This is especially relevant if one considers that Twitter has no ways of indexing tweet contents, and that the only available categorization "mechanism" is the #hashtag, which is totally dependent of a user's will to use it. This paper presents an intelligent platform and framework, named MISNIS, that facilitates these issues and allows a non-technical user to easily mine a given topic from a very large tweet's corpus and obtain relevant contents and indicators such as user influence or sentiment analysis.

Keywords: Twitter; Intelligent Topic Mining; Fuzzy Fingerprints; MongoDB; MISNIS.

1 Introduction

When Twitter was launched in 2006 as a simple public social networking service enabling users to send and read short 140-character messages, hardly anyone could predict that it would become a major tool for spreading news, for dissemination of positions or ideas, and for the commenting and analysis of current world events. It all started with the "Arab Spring", in 2010, where Twitter was used as an alternative means of communicating to the outside world what was censored by state controlled traditional news broadcasters. During the subsequent years, events such as the "Spanish protests", the "London riots" or the "Taksim Gezi Park protests", further increased the notion that important events are often commented in Twitter before they become "public news". This has led to a change in how the public perceives the importance of social networks, and even news agencies and networks had to adapt and are now using Twitter as a potential (and some times preferential) source of information.

Sankaranarayanan (2009), showed how Twitter can be used to automatically obtain breaking news from the tweets posted by users, and exemplifies that when Michael Jackson passed away, "the first tweet was posted 20 minutes after the 911 call, which was almost an hour before the conventional news media first reported on his condition". This further enforces the importance of automatically analyzing the massive amount of information on Twitter.

However, using Twitter as a source of information involves many technical obstacles, of which the first is collecting and dealing with the amount of flowing information. As of mid 2015, more than 500 millions tweets covering thousands of different topics are published daily (almost 6000 tweets per second!). Collecting, storing, managing and visualizing such large amounts of information is a far from trivial problem and demands dedicated and intelligent hardware and software platforms.

Even assuming one is able to access all tweeted contents, there is still the problem of filtering which content is relevant for a given topic of interest. This is far from trivial, even if we simply consider doing it on a daily basis: on a given day it is very unlikely that more than a few thousand tweets are relevant to a given discussion topic (even when considering major topics). We are talking about detecting 0.001%-0.01% of the 500 million daily tweets, which is basically trying to find a needle on a hay stack. Twitter contributes to solve this problem by providing a list of top trends (Twitter, 2010) and the hashtag # mechanism: when referring to a certain topic, users are encouraged to indicate it through the use of a hashtag. E.g., "#refugeeswelcome in Europe!" indicates the topic of the tweet is the current refugees crisis in Europe. However, not all tweets related to a given topic are hashtagged. In fact, only roughly 16% of all tweets are hashtagged (Mazzia, 2010). These numbers have been confirmed by our experiments, and can be partially explained by the fact that 140 characters is hardly enough to communicate a thought, and including an #hashtag further aggravates the lack of available space. It is therefore clear that, in order to properly analyze a given discussion topic, it is essential to retrieve as much of the remaining 84% untagged information as possible. Since no other tagging mechanisms exist in Twitter, the process of retrieving tweets that are related to a given topic, our needle on a haystack, must use some kind of text classification process in order to detect if the contents of a given tweet is somehow related with the intended topic. This classification process must simultaneously be able to retrieve only the relevant tweets (i.e., have high values of precision and recall), and to be computationally efficient in order to deal with the huge amount of data.

Additional tasks of interest when considering the use of Twitter as an information source, include finding which of the topic related tweets have more relevance (e.g. by finding who are most important "actors" discussing the topic), performing sentiment analysis, extract statistics on the topic origin and respective spatio-temporal evolution, etc.

In this article, we present an intelligent platform, MISNIS, which addresses the issues mentioned above, and can be used as an expert system by social scientists when studying social networks' impact in society. The platform can be divided into two major blocks (Figure **1**): 1) Smart mechanisms for collecting, storing and managing Twitter information; 2) Intelligent mechanisms to retrieve, analyze and represent the information that is relevant for a given topic.

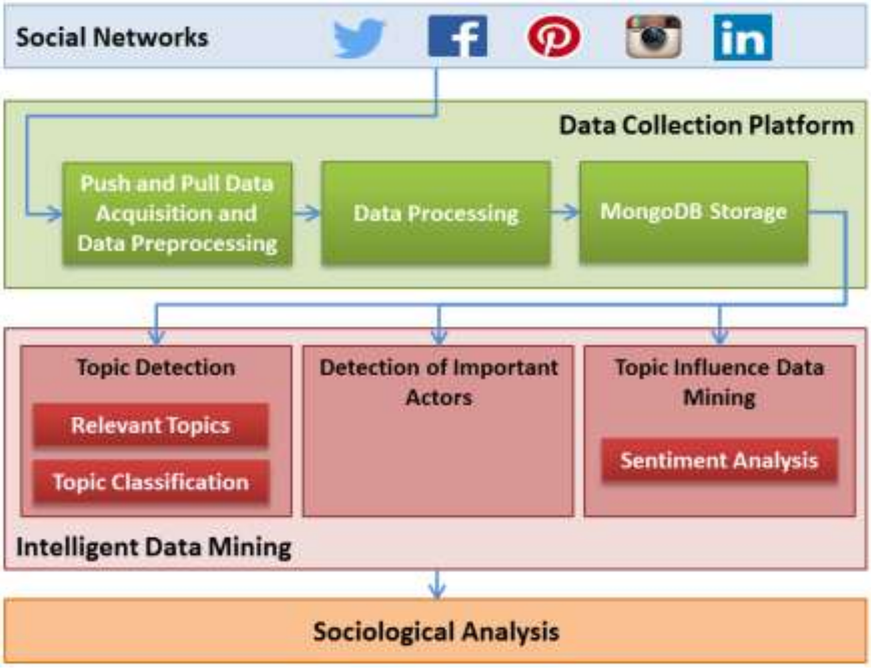


Figure 1: MISNIS Framework architecture

The paper is organized as follows: In section 2 we describe some related work relevant to the developed platform; Section 3 describes the architecture of the framework that was implemented for Twitter data acquisition, storage, management and visualization; Section 4 focuses on the expert system for intelligent Twitter data mining added to the framework; Section 5 presents a small case study used to exemplify the developed platform and framework; Finally, Section 6 presents some conclusions.

2 Related Work

2.1 Large Scale Social Data Acquisition and Storage

The analysis of the content and information shared on social networks has been proved useful in various fields, including Politics, Marketing, Tourism, Public Health, and Safety. Twitter is amongst the most widely used social networks, making available about 500 million tweets every day[1], on average. Twitter provides free access to part of the information produced by its users through public APIs (*Application Programming Interface*), and the popularity of Twitter as a source of information has led to the development of numerous applications and to new research methods in various fields. For example, Paul *et al.* (2011) developed a method for tracking disease risk factors by measuring the behavioral level, tracking diseases by geographic regions to analyze the symptoms and medication applied. The study was based on about 1.5 Million tweets related to health that contained references to various ailments including allergies, obesity and insomnia. Santos *et al.* (2013, 2014) used a set of approximately 2700 tweets produced in Portugal to

[1] https://about.twitter.com/company (accessed in 14-05-2015).

predict the incidence and spread of the *influenza* virus through the Portuguese population. Widener *et al.* (2014) used information extraction and sentiment analysis (through a data mining framework), to try to understand how geolocated tweets can be used to research the prevalence of healthy and unhealthy food in contiguous regions of the United States. Other studies related to public health were also reported by Culotta (2010) and by Scanfeld *et al.* (2010). Twitter was also used as a source of information to help identifying or locating the occurrence of earthquakes, taking into account that "when an earthquake occurs people produce many *posts* on Twitter related to the event, which permits the identification of earthquakes simply by observing the increase in the tweet volume" (Sakaki *et al.* 2010). Kumar *et al.* (2013b) proposes an approach to identify a subset of users and their location to justify them to be followed in disaster situations in order to get a quick access to useful information about the event. During a crisis a particular user's location is an important factor in determining whether he/she is likely to publish relevant information on the state of crisis. For instance, in the eventuality of an earthquake, tweets produced in a place close to the earthquake are likely to be more relevant to assess the situation than tweets produced from a more distant location. Other studies have been produced based on similar topics (Mendoza *et al.* 2010, Qu *et al.* 2011, Lachlan *et al.* 2014). Gerber (2014) tried to predict criminal activity in the largest city of the United States of America using tweets marked in space and time. Since tweets are public, officially made available by Twitter services, the development of linguistic analysis models that enable the automatic identification of topics related to the commission of a crime may be considered quite relevant not only in preventing similar crimes but also to support decision making under trial in court of law.

The design of software architectures for capturing Twitter information and extracting relevant knowledge from tweets represents a major challenge, not only due to the massive amounts of streamed data, but also due to possible access limits imposed by Twitter (Oussalah *et al.* 2013). Most of the work reported in literature restricts the data in some way, in order to respect the limits imposed by Twitter. For example, Perera *et al.* (2010) describes a software architecture based on the Twitter API based on Python and MySQL that collects tweets sent only to specific users. Twython, a Python wrapper, was used to obtain spatial data (location, name, description, etc.) of the authors of the tweets. The collecting process runs over in 5-minute intervals and the collected tweets sent to a particular Twitter *id* user, including the President Barack Obama. Anderson *et al.* (2011) reports a concurrency-based software architecture that allows collecting a large volume of data, with a theoretical maximum of 500 million tweets per day. The developed code is multi-threaded and therefore adapted to run on machines with multiple processors, uses Spring, MVC, Hibernate and JPA *frameworks*, and the infrastructure components: Tomcat, Lucene (for tweet indexation), MySQL (to store the collected data). Marcus *et al.* (2011) developed *TwitInfo*, a platform for collecting and processing tweets in real time for sentiment analysis. The architecture proposed by Oussalah *et al.* (2013) collects tweets continuously and in real time using the Streaming API, aiming at semantically and spatially analyze the collected data. The collected tweets are restricted to a rectangular area bounded by geographical coordinates (longitude and latitude), and the software implementation is based on Django (*framework* for developing web applications in Python), Lucene, and MySQL. This architecture allows searching for tweets using the text, username or location.

2.2 Tweet Topic Detection

One of the goals of this work is essentially to automatically classify tweets into a set of topics of interest. Tweet Topic Detection involves deciding if a given tweet is related to a given, usually #hashtagged, topic. Basically this can be categorized as a classification problem, albeit one with some particular characteristics that need to be addressed specifically: (1) it is a text classification problem, with an unknown and large number of categories, where the texts to be classified are very short texts (up to 140 characters); (2) it fits the Big Data paradigm due to the huge amounts of streaming data.

We distinguish between Topic Classification and Topic Detection. The former is broadly known in Natural Language Processing (NLP) as Text Categorization, and consists of finding the correct topic (or topics) for each document, given a restricted set of categories (subjects, topics) such as politics, sports, music, etc., and a collection of text documents (Feldman, 2006), in this case, tweets; the tweets will often belong to at least one of those categories and it is very rare that a tweet does not fit into any topic. The latter takes on a more detailed approach, where an attempt is made to determine the topic of the document, given a predetermined large set of possible topics, where the topics are so unique amongst themselves that there is a high probability that a tweet without a hashtag may very well not belong to any of the current trends.

When considering this difference, the most similar works on topic detection within Twitter are those related with emerging topics or trends, as for example the works of Mathiodakis (2010), Cataldi (2010) Kasiviswanathan (2011) or Saha (2012). In these works the authors use a wide variety of techniques regarding text analysis to find the most common related words and hence detect topics. For the case of topic detection, we assume the existence of trending topics and we aim at efficiently detecting tweets that are related to them, despite not being explicitly marked as so.

It is also possible to find several works regarding Topic Classification. In Lee (2011) an attempt is made to classify Twitter Trending Topics into 18 broad categories, such as: sports, politics, technology, etc., and their experiments on a database of randomly selected 768 trending topics (over 18 classes) show that, using text-based and network-based classification modeling, a classification accuracy up to 65% and 70% can be achieved, respectively.

In what concerns what we define as topic detection, the most promising method is Twitter Topic Fuzzy Fingerprinting (Rosa et all, 2014a, 2014b). Fingerprint identification is a well-known and widely documented technique in forensic sciences. In computer sciences a fingerprint is a procedure that maps an arbitrarily large data item (such as a computer file, or author set of texts) to a much compact information block, its fingerprint, that uniquely identifies the original data for all practical purposes, just as human fingerprints uniquely identify people. Fuzzy Fingerprints were originally introduced as a tool for text classification by Homem and Carvalho (2011). They were successfully used to detect authorship of newspaper articles (out of 73 different authors). For text classification purposes, a set of texts associated with a given class is used to build the class fingerprint. Each word in each text represents a distinctive event in the process of building the class fingerprint, and distinct word frequencies are used as a proxy for the class associated with a specific text. The set of the fuzzy fingerprints of all classes is known as the fingerprint library. Given a fingerprint library and a text to be classified, the text fingerprint is obtained using a process similar to the one used to create the fingerprint of each class, and then a similarity function is used to fit the text into the class that has the most similar fingerprint.

In order to use Fuzzy Fingerprints for tweet topic detection, several procedural changes were proposed by Rosa *et. al* (Rosa, 2014a, 2014b). According to the authors, the Twitter Topic Fuzzy Fingerprints performed very well on a set of 2 millions English, Spanish and Portuguese tweets collected over a single day, beating other widely used text classification techniques. The training set consisted of 11000 tweets containing the 22 of the top daily trends (hashtagged topics). 350 unhashtagged test tweets were properly classified with an f-measure score of 0.844 (precision=0.804, recall=0.889). Further work by Rosa (2014), used a training set of 21000 tweets, from “21 impartially chosen topics of interest out of the top trends of the 18th of May, 2013”. The test set was made of “585 tweets that do not contain any of the top trending hashtags” and “each tweet was impartially annotated to belong to one of the 21 chosen top trends”. After extensive parameter optimization using a development set, the fuzzy fingerprint method scored an f-measure of 0.833 proving to be not only more accurate than other well known classifying techniques (kNN and SVM), but also much faster (177 times faster than kNN and 419 times faster than SVM).

2.3 User Influence

The concept of influence is of much interest for several fields, such as sociology, marketing and politics. Empirically speaking, an influential person can be described as someone with the ability to change the opinion of many, in order to reflect his own. While Rogers (1982) supports this statement, claiming that “a minority of users, called influentials, excel in persuading others?”, more modern approaches (Domingos, 2001) seem to emphasize the importance of interpersonal relationships amongst ordinary users, reinforcing that people make choices based on the opinions of their peers. The point is that “influence” is an abstract concept, which makes it exceptionally hard to quantify.

Several studies have attempted to accomplish this goal. In (Cha, 2010), three measures of influence were taken into account, regarding Twitter: in-degree, re-tweets and mentions, where “in-degree is the number of people who follow a user; re-tweets mean the number of times others forward a user's tweet; and mentions mean the number of times others mention a user's name”. It concluded that while in-degree measure is useful to identify users who get a lot of attention, it “is not related to other important notions of influence such as engaging audience”. Instead “it is more influential to have an active audience who re-tweets or mentions the user”.

In (Leavitt, 2009), the authors conclude that within Twitter, “news outlets, regardless of follower count, influence large amounts of followers to republish their content to other users", while "celebrities with higher follower totals foster more conversation than provide retweetable content".

InfluenceTracker (Razis, 2014) is a framework that rates the impact of a Twitter account taking into consideration an Influence Metric, based on the ratio between the number of followers of a user and the users it follows, and the amount of recent activity of a given account. It also calculates a Tweet Transmission rate where the "most important factor (...) is the followers' probability of re-tweeting". Cha (2010) also shows "that the number of followers a user has, is not sufficient to guarantee the maximum diffusion of information in Twitter (...) because, these followers should not only be active Twitter users, but also have impact on the network".

Even if one agrees on the measures that best represent influence, aggregating and computing the measures is not a trivial task since user interactions should not be ignored. A sound approach consists in using graphs and to compute user relevance recurring to graph centrality algorithms.

In graph theory and network analysis, the concept of centrality refers to the identification of the most important vertices within a graph, in this case, the most important users. We therefore define a graph G(V,E) where V is the set of users and E is the set of directed links between them.

Currently most "famous" centrality algorithm is PageRank (Page, 1998, 1999). It is one of Google's search engine methods, with web pages used as nodes and back-links forming the edges of the graph. The PageRank is considered to be a random walk model, because the weight of a page/node is "the probability that a random walker (which continues to follow arbitrary links to move from page to page) will be a node at at any given time". A damping factor is used as the "probability of the random walk to jump to an arbitrary page, rather than to follow a link, on the Web" and is "required to reduce the effects on the PageRank computation of loops and dangling links in the Web" (Phuoc, 2009).

Other less complex and with guaranteed convergence centrality methods exist, such as for example, Katz (Katz, 1953), which are often preferred over PageRank for that reason.

2.4 Sentiment Analysis

Sentiment Analysis is a relevant and well-known task that consists of extracting sentiments and emotions expressed in texts. Being the first step towards the online reputation analysis, it is now gaining particular relevance because of the rise of social media, such as blogs and social networks. The increasing amount of user-generated contents constitute huge volumes of opinionated texts all over the web that are precious sources of information, especially for decision support. Sentiment Analysis can be used to know what people think about a product, a company, an event, or a political candidate. Sentiment analysis can be performed at different complexity levels, where the most basic one consists just on deciding whether a portion of text contains a positive or a negative sentiment.

Dealing with the huge amounts of data available on Twitter demand clever strategies. One approach combines sentiment analysis and causal rule discovery (Dehkharghani, 2014). Other, by Kontopoulos (2013) uses ontologies. An interesting and simpler idea, explored by Go (2009), consists of using emoticons, abundantly available on tweets, to automatically label the data and then use such data to train machine learning algorithms. The paper shows that machine learning algorithms trained with such approach achieve above 80% accuracy, when classifying messages as positive or negative. A similar idea was previously explored by Pang (2002) for movie reviews, by using star ratings as polarity signals in their training data. This latter paper analyses the performance of different classifiers on movie reviews, and presents a number of techniques that were used by many authors and served as baseline for posterior studies. As an example, they have adapted a technique, introduced by Das and Chen (2001), for modeling the contextual effect of negation, adding the prefix `NOT_` to every word between a "negation word" and the first punctuation mark following the negation word.

Common approaches to sentiment analysis also involve the use of sentiment lexicons of positive and negative words or expressions (Stone, 1996; Hu, 2004; Wilson, 2005; Baccianella, 2010). Another research approach involves learning polarity lexicons and can be specially useful for dealing with large corpora. The process starts with a seed set of words and the idea is to increasingly find words or phrases with similar polarity, in semi-supervised fashion (Turney, 2002). The final lexicon contains much more words, possibly learning domain-specific information, and therefore is more prone to be robust. The work reported by Kim (2004) is another example of learning algorithm that uses WordNet synonyms and antonyms to learn polarity.

3 Twitter Data Acquisition, Storage, Management and Visualization

In order to circumvent Twitter data access restrictions and build a tweets repository from where data could be efficiently managed, intelligently retrieved and visualized, we developed an information system consisting of four main modules (Figure **2**): 1) Data collection; 2) Data expansion; 3) Data access; 4) Visualization.

The presented system focus mainly on Portuguese Twitter data (Tweets produced in Portugal and in European Portuguese), but can easily be adapted and expanded for most countries and languages.

In module Collect we retrieve geolocated tweets produced in Portugal and discard those that are not recognized as written in European Portuguese. The collection uses Twitter StremingAPI, which implies that at most 1% of the data is collected. All collected tweets are stored in MongoDB[2]. In module Expand we identify each individual previously collected user, and explore their timelines to retrieve past tweets (and add them to the MongoDB). In module Access we implemented a REST API[3] as an abstraction to intelligently access the database. Module Visualization implements a dashboard to visualize metrics, indicators and any queried information. Each module is detailed in the following sections.

[2] https://www.mongodb.org, last accessed, July 2015

[3] http://www.restapitutorial.com/, lasta accessed, July 2015

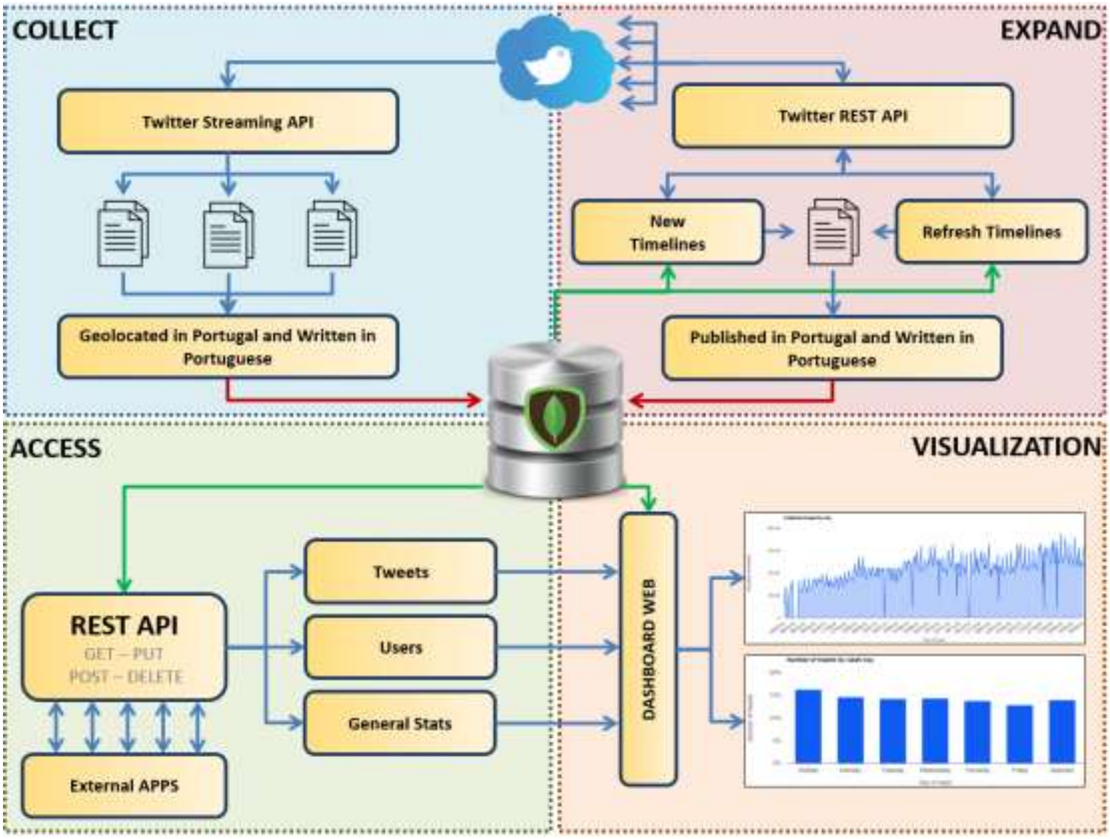


Figure 2: Architecture of the Twitter data collection, management and visualization system.

3.1 Geolocated Data Collection

The Twitter Streaming API *statuses/filter* allows the access to tweets being tweeted at the time of request. Several filters can be used to tailor the requested information: the API allows filtering by keywords, hashtags, user ID and geographically delimited regions (Kumar 2013a). The number of parameters and the volume of returned information is limited by Twitter. Currently each request allows for a maximum of 400 keywords, 25 geographical regions or 5000 user IDs, and up to a maximum of 1% of all currently flowing tweets are searched for and returned.

The geolocation of a tweet can be obtained using two different processes: i) directly from the tweet when the user opts to make his location known at time of publishing; ii) using the information contained in the user profile *location* field. The percentage of geolocated tweets is low, under one fifth of all tweets[4], and as such, it is easier to comply with Twitter API restrictions when filtering for geolocated tweets:

- Taking into consideration that currently there are 500 million tweets per day[5], it is theoretically possible to retrieve up to 5 million tweets per day using the Streaming API from a single user account;

[4] https://pressroom.usc.edu/twitter-and-privacy-nearly-one-in-five-tweets-divulge-user-location-through-geotagging-or-metadata/, last accessed July 2015.
[5] https://about.twitter.com, last accessed October 2015.

- Previous works (Brogueira, 2014a, 2014b) have shown that in 2014, 60000 geolocated tweets were produced per day, in Portugal and using (European) Portuguese, i.e., almost 2 orders of magnitude lower than the 5 million Streaming API daily limits.

The implemented platform works on a 24 hours per day basis using the Twitter Streaming API to collect geolocated flowing tweets within Portugal. The delimitation of the geographical area of mainland Portugal and the archipelagos of Madeira and Azores uses the Twitter REST API *geo/search*. Note that this area includes small parts of Spain and North Africa that are filtered *a posteriori*.

The collected tweets are compressed and stored on a hard drive (in as "real time" as possible), and later filtered and stored in a MongoDB database. The filtering operation consists of considering only tweets that are produced in Portugal (field *place.country = "pt"*) and written in European Portuguese (field *lang = "pt"*). The doublecheck allows for the detection of tweets where the Twitter language detection algorithm[6] fails, which is not uncommon between Portuguese, Spanish and Galician.

In addition to the tweet filter and storage operation, the author of each tweet is identified using the field *user.id*, and if he is unknown, added to a "Users" MongoDB collection. For each new user it is created a JSON[7] document containing: i) the *user.id*; ii) the date of first detection; iii) control flags (see section 3.2).

3.2 Database Expansion

The tweet corpus expansion is based on the retrieval of the timeline of each user present in the "Users" collection. The user's timeline is the record of the user's recent Twitter activity, and can be accessed via the REST API *statuses/user_timeline*.

The API returns the most recent 3200 tweets of a given user. However, Twitter imposes several restrictions that hinder the timeline collection process: up to 180 requests are authorized per 15 minute period and per authenticated Twitter API access account (Kumar, 2013a), and each request returns at most 200 tweets. Therefore, 16 API requests are needed in order to retrieve a 3200 tweets timeline. As such, it is only possible to retrieve the timeline of 11 different users per 15-minute period (around 1080 timelines per day) when using a single Twitter account.

Since an average of 232 new users are identified per day by the system (Brogueira, 2015a), and the total of registered users is, by August 2015, above 120K, collecting the timelines of every single new user, and updating existing users' timelines on a continuous basis using a single Twitter API access account is an increasingly lengthy process, that is not viable and far from straightforward due to the above-mentioned restrictions.

Timeline retrieval in the developed system uses 15 different Twitter API access accounts that are synchronized and optimized to prevent repeated invocations and

[6] https://blog.twitter.com/2013/introducing-new-metadata-for-tweets, last accessed July 2015.
[7] http://json.org/, last accessed June 2015.

failures due to exceeding the API limits during the 15-minute window. The retrieval process considers three different scenarios:

i) Integration of a new user, which implies obtaining the complete timeline (up to 3200 tweets);

ii) Existing user, for whom it is necessary to retrieve tweets produced since the date of the last retrieved tweet;

iii) Blocked access users, i.e., users that have explicitly blocked the access to their timelines.

Each account is dynamically assigned to one of the first two scenarios taking into consideration the amount of timelines to process for each. Blocked users are kept out of the loop and checked sporadically in order to detect eventual status change.

Each account is also associated to a JSON document containing the strings used for Twitter API authentication (OAuth), and also flags that identify to which scenario the account is currently assigned to and how it should operate. More details can be found in (Brogueira, 2015b).

3.3 Data Access and Data sharing

The data access module consists of a REST API developed in order to facilitate the access to the information stored in the database, allow access by third party applications, and enable the developed Dashboard (see section 3.4).

REST stands for Representational State Transfer, a set of constraints and principles used in web interface architectures. A REST API is a set of data and functions that facilitate information exchange between applications and web services designed according to the REST principles.

The developed REST API makes a set of endpoints available for interaction with the MongoDB. Table 1 presents some of the endpoints developed for information access. They are divided into three categories: i) Access to tweets information; ii) Access to user information; iii) Access to previously processed statistics (performed on the stored data). Table 2 presents endpoints available for adding information to the database, namely information resulting from the intelligent data processing methods presented in Section 4.

Table 1: Database access endpoints of the REST API.

Endpoint	**Parameters**	**Return Data**
/api/<collection>/page/	<String: page_id>	All information concerning a specific Tweet
/api/<collection>/day/	<int: day>	Set of 1K tweets ordered by decreasing publishing date
/api/<collection>/user/	<String: screen_name>	Tweets produced by day
/api/<collection>/query/	<String: query>	All tweets published by user
/api/user/firstProfile/	<String: screen_name>	Set of tweets according to the filter specified in parameter ”query”
/api/user/lastProfile/	<String: screen_name>	User profile when first tweet included in the database was published
/api/user/ageGender/	<String: screen_name>	User profile when his last tweet included in the database was published
/api/geolocated/day/		Fields of user profile used to infer age and gender
/api/geolocated/hour/		Number of new users identified per day
/api/geolocated/weekDay/		Geolocated tweets collected per hour
/api/timeline/day/		Geolocated tweets collected per week day
/api/users/day/		Tweets collected per day

Table 2: REST API endpoints for saving “intelligent analysis” results into MongoDB.

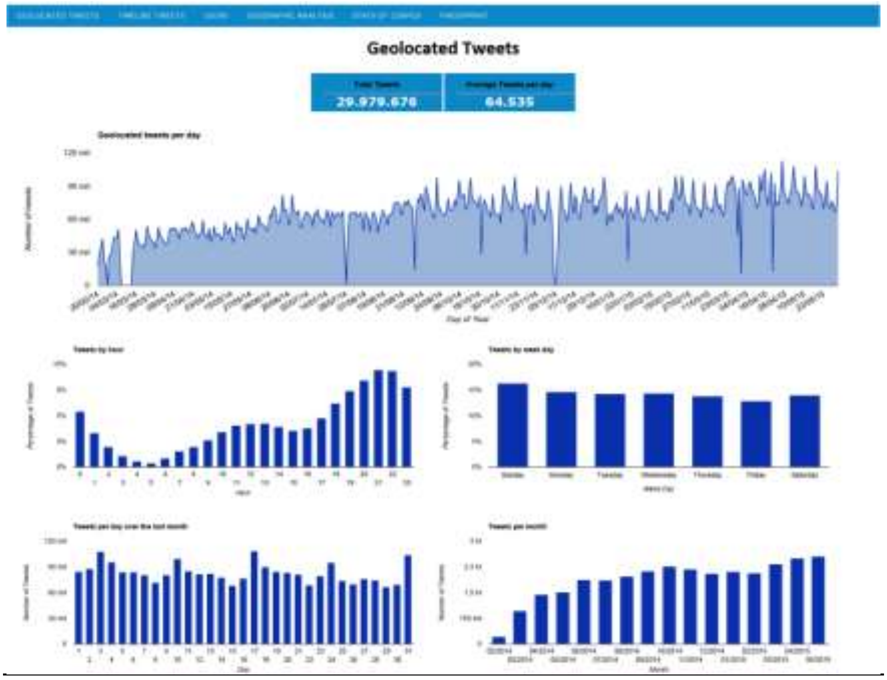
Endpoint	**Description**
/api/t2f2tweets/	Tweets related to a given topic (processed using Twitter Topic Detection)
/api/popusers/	Most relevant users on a given topic of discussion

3.4 Data Visualization

The Data Visualization module consists on a web dashboard integrating several data indicators. The dashboard is implemented using the Google Charts API[8] and the REST API presented in the previous section. The dashboard includes charts and statistics for geolocated tweets, timeline tweets and user information.

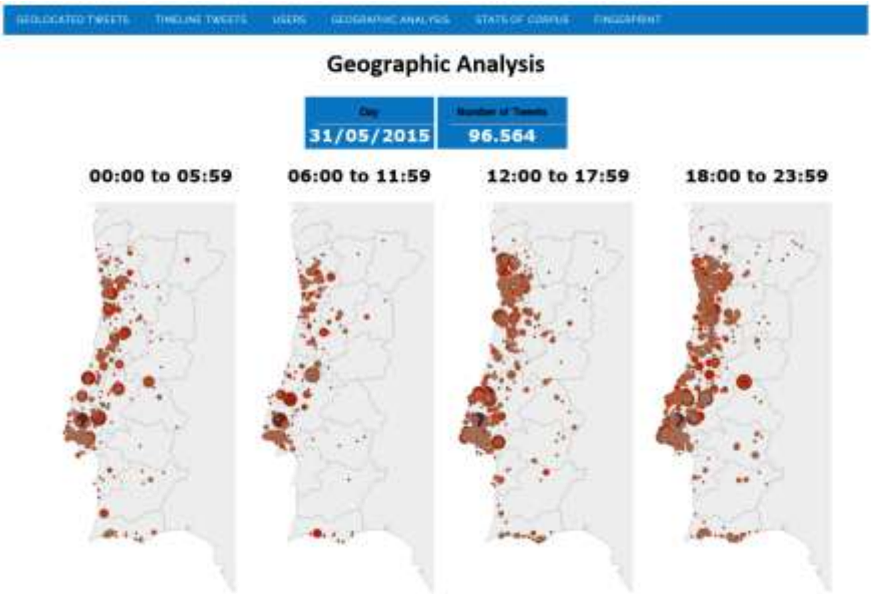
Since some of the statistics and charts involve a large volume of data, and the processing of the respective queries is not feasible in real time, such queries are therefore pre-processed automatically on a daily basis. In such cases, the visualized information refers to data collected up to the previous day. Figure **3** shows an example of such queries, where information concerning geolocated tweets is visualized.

It is also possible to query and visualize in “real time”. For example, Figure **4** shows the dashboard for the location of geolocalized tweets collected on a given date.



[8] https://developers.google.com/chart/?csw=1

Figure 3: Dashboard containing indicators about geolocated tweets



Geographic Analysis

Day	Number of Tweets
31/05/2015	96.564

Figure 4: Visualization of collected geolocated tweets on a given day

4 Intelligent twitter topic mining add-on

The framework described in the previous section allows the collection, management and visualization of an extensive tweet corpus. In this section we address how we add to the framework the capabilities to intelligently retrieve relevant information from the stored corpus. Despite the underneath complexity, the process is designed to be easily accessible to users lacking technical knowledge, such as social scientists or sociologists. The process is represented in Figure **5**, succinctly described as follows, and detailed in the next subsections.

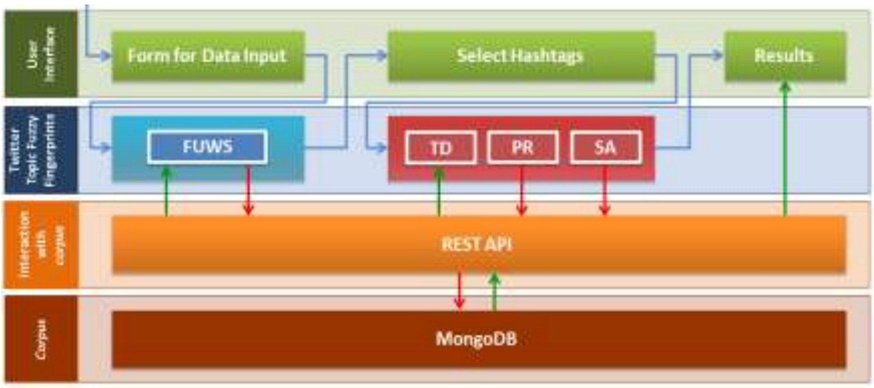


Figure 5: Intelligent Twitter topic mining

All interaction with the user occurs at a “User interface” layer. A “Twitter Topic Fuzzy Fingerprints” layer is used to process user inputs, interact with the REST API, process the data returned from MongoDB and present the results to the user.

When a user wants to retrieve information relevant to a given topic of interest, it must input the search period (begin date, end date) and any keywords/#hashtags using the available web interface (Figure **6**-1).

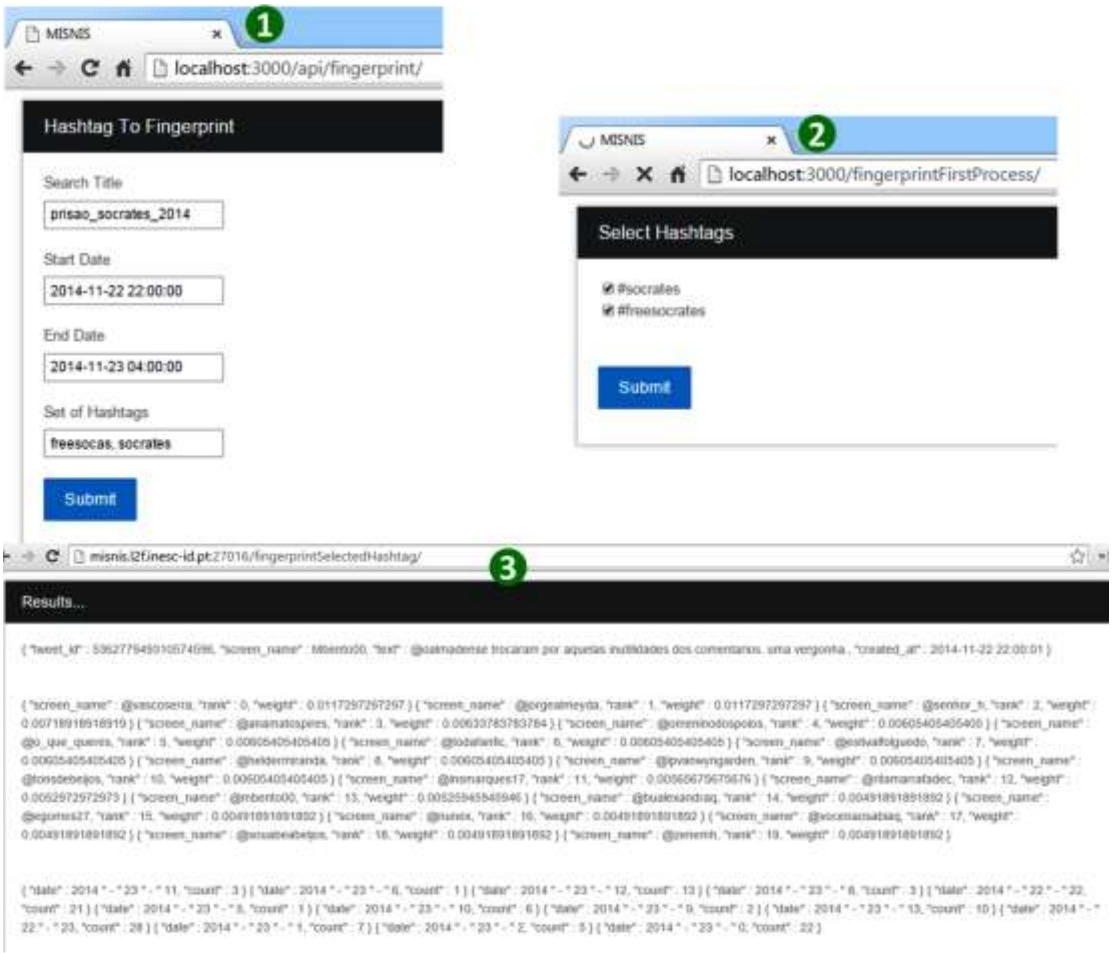
The system will then use a fuzzy word similarity algorithm (FUWS – see section 4.2) to search the MongoDB for similar keywords and #hashtags found in the database during the referenced time period. (Figure **6**-2). As a result of this step, all tweets containing such keywords/#hashtags are retrieved and stored in a temporary collection, and an output list of all similar keywords/#hashtags is returned and shown to the user.

The user can prune the list and indicate which of the returned keywords and hashtags might or might not be useful within the context of the topic to be analyzed.

The pruned keywords/hastahgs list is then passed back to the Twitter Topic Fuzzy Fingerprints layer, where it is used to create a topic Fuzzy Fingerprint (see section 4.1) based on the tweets stored in the temporary collection. This fingerprint is then used to find tweets in MongoDB that are related to the topic of interest (Figure **6**-3). The set of relevant tweets is written back into a separate collection in MongoDB. Tweets are considered relevant to the topic if they match the fingerprint to a given degree. It should be noted that the method can find relevant tweets even when they do not contain any of the items present in the pruned list of keywords and/or #hashtags. A user only needs to provide a single relevant #hashtag since the method is able to create the topic fingerprint based on contents of the tweets found after applying the FUWS.

The relevant tweets are then processed in order to find the top-20 most influential users in propagating the topic in study, (see section 4.3), and sentiment analysis is performed (see section 4.4). The resulting data is also written back into MongoDB in separate collections.

A Results webpage (Figure **7**), containing relevant information and automatically obtained by querying the above-mentioned collections, is presented to the user as a result of the process. The collections can also be queried using any of the developed REST API commands (in a user friendly way).



1

localhost:3000/api/fingerprint/

Hashtag To Fingerprint

Search Title: prisao_socrates_2014

Start Date: 2014-11-22 22:00:00

End Date: 2014-11-23 04:00:00

Set of Hashtags: freesocas, socrates

2

localhost:3000/fingerprintFirstProcess/

Select Hashtags

☑ #socrates
☑ #freesocrates

3

misnis.l2f.inesc-id.pt:27016/fingerprintSelectedHashtag/

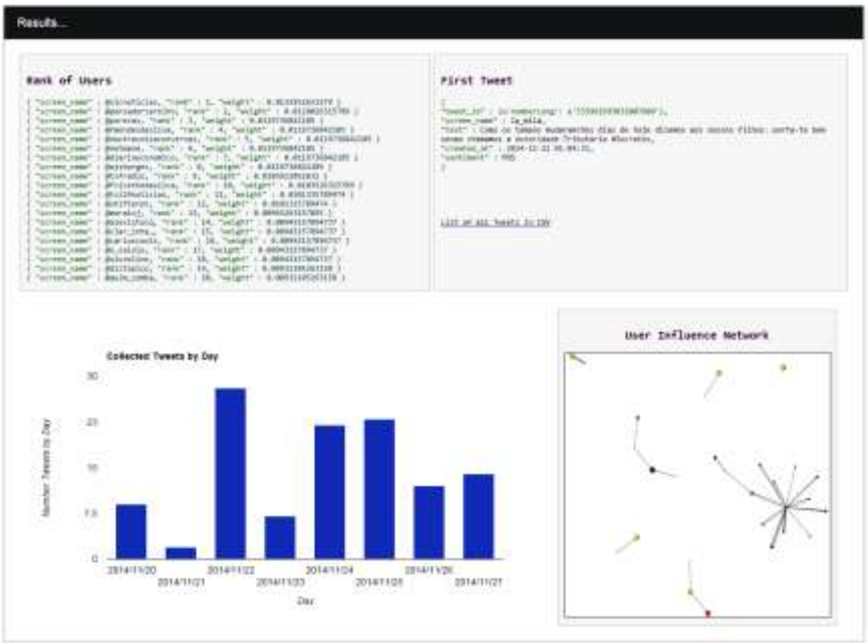
Results...

```
{ "tweet_id" : 536277945910574096, "screen_name" : Mbento00, "text" : @dalmadense trocaram por aquelas inutilidades dos comentarios. uma vergonha , "created_at" : 2014-11-22 22:00:01 }

{ "screen_name" : @vascoserra, "rank" : 0, "weight" : 0.0117297297297 } { "screen_name" : @jorgealmeyda, "rank" : 1, "weight" : 0.0117297297297 } { "screen_name" : @senhor_h, "rank" : 2, "weight" :
0.00718918918919 } { "screen_name" : @anamatospires, "rank" : 3, "weight" : 0.00633783783784 } { "screen_name" : @omeninodospobos, "rank" : 4, "weight" : 0.00605405405406 } { "screen_name" :
@o_que_queres, "rank" : 5, "weight" : 0.00605405405406 } { "screen_name" : @todafanfic, "rank" : 6, "weight" : 0.00605405405405 } { "screen_name" : @estwaffolguedo, "rank" : 7, "weight" :
0.00605405405406 } { "screen_name" : @heldermiranda, "rank" : 8, "weight" : 0.00605405405405 } { "screen_name" : @ipvanwyngarden, "rank" : 9, "weight" : 0.00605405405405 } { "screen_name" :
@fonsdebeijos, "rank" : 10, "weight" : 0.00605405405405 } { "screen_name" : @insmarques17, "rank" : 11, "weight" : 0.00566756756676 } { "screen_name" : @ritamariafadec, "rank" : 12, "weight" :
0.00529729729730 } { "screen_name" : @mbento00, "rank" : 13, "weight" : 0.00525945945946 } { "screen_name" : @bualexandraq, "rank" : 14, "weight" : 0.00491891891892 } { "screen_name" :
@ejgomes27, "rank" : 15, "weight" : 0.00491891891892 } { "screen_name" : @tunex, "rank" : 16, "weight" : 0.00491891891892 } { "screen_name" : @vocemamabag, "rank" : 17, "weight" :
0.00491891891892 } { "screen_name" : @souabeabeijos, "rank" : 18, "weight" : 0.00491891891892 } { "screen_name" : @jenemh, "rank" : 19, "weight" : 0.00491891891892 }

{ "date" : 2014 " - " 23 " - " 11, "count" : 3 } { "date" : 2014 " - " 23 " - " 6, "count" : 1 } { "date" : 2014 " - " 23 " - " 12, "count" : 13 } { "date" : 2014 " - " 23 " - " 8, "count" : 3 } { "date" : 2014 " - " 22 " - " 22,
"count" : 21 } { "date" : 2014 " - " 23 " - " 5, "count" : 1 } { "date" : 2014 " - " 23 " - " 10, "count" : 6 } { "date" : 2014 " - " 23 " - " 9, "count" : 2 } { "date" : 2014 " - " 23 " - " 13, "count" : 10 } { "date" : 2014 " - "
22 " - " 23, "count" : 28 } { "date" : 2014 " - " 23 " - " 1, "count" : 7 } { "date" : 2014 " - " 23 " - " 2, "count" : 5 } { "date" : 2014 " - " 23 " - " 0, "count" : 22 }
```

Figure 6: Intelligent topic mining steps when viewed from the user dashboard



Results...

Rank of Users

[illegible]

First Tweet

[illegible]

List of all tweets in CSV

Figure 7: Intelligent topic mining results' visualization

4.1 Twitter Topic Fuzzy Fingerprints

In order to perform tweet topic detection, we used the Twitter Topic Fuzzy Fingerprints method (Rosa 2014a, 2014b, 2014c), which adapts the original Fuzzy Fingerprints method to the characteristics of individual tweets.

As described in section 2.2, textual fuzzy fingerprinting works by comparing the similarity of the fuzzy fingerprint of an individual text, with the fingerprint of a possible author (created based on a set of texts written by that author). When using Fuzzy Fingerprints to detect if a tweet is related to a given topic, we start by creating the fingerprint of the topic (instead of an author), plus the fingerprint of a set of trending topics.

A topic fingerprint is created based on a set of training tweets known to be related to the topic in question. All tweets containing the #hashtag of the topic are retrieved from the database, and their text is preprocessed to remove unimportant information, such as words with less than 3 characters. The next step consists in obtaining the word frequency in the processed tweets. Only the top-*k* words are considered for the fingerprint. The computation of the top-k words and respective frequency is done using an approximated counting method, FSS – Filtered Space Saving (Homem, 2010) for efficiency reasons. This process is repeated for all the trending topics and the topic to be detected.

The next step differs from the original method: we account for the Inverse Class Frequency of each word, icf_v (1), which is an adaptation of the well-known Inverse Document Frequency (*idf*), to reorder the top-*k* word lists of all topics.

$$icf_v = log\frac{J}{J_v}, \tag{1}$$

In (1), J is cardinality of all considered topics, and J_v is the number of topics where word v occurs.

The product of the frequency of each word v with its icf_v, is used to distinguish the occurrence of common words and get a new ordered k-sized rank for each #hashtagged topic.

The next step consists in fuzzifying each top-k list in order to obtain the fingerprint for each #hashtagged topic. A membership value is assigned to each word of the top-k list based on its order (instead of its frequency or its icf). In MISNIS we used the fuzzifying function represented in (2), which is inspired by the Pareto rule, where roughly 80% of the membership value is assigned to the top 20% words.

$$\mu(i) = \begin{cases} 1-(1-b)\frac{i}{kb}\ , \ i \leq a \\ \frac{a}{k}\left(1-\frac{i-a}{k-a}\right), i > a \end{cases} , a, b = 0.2 \tag{2}$$

The fingerprint (Φ_j) is a size-k fuzzy vector, where each position j contains a word v_i and a membership value $\mu(i)$ representing the fuzzified value of the rank of v_i obtained by the application of (2). The set of all computed topic fingerprints constitutes the fingerprint library.

Once the fingerprint library is created, it is possible to look in MongoDB for tweets that despite not containing an intended #hashtag, discuss the same topic. In the original fingerprint method, this would be done by computing the fingerprint of an individual tweet and comparing it to the topic fingerprint. However, knowing that a text fingerprint is essentially based on the fuzzification of the order of the word frequency of the text, and that tweets have a maximum of 140 characters, it is not possible (or useful) to create the fingerprint on an individual tweet, since within a tweet, very few relevant words (if any!), are repeated. As such, a new similarity score, the Tweet to Topic Similarity Score (T2S2) was developed to test for the similarity between a given tweet and a topic fingerprint. The T2S2 score (3) does not take into account the size of the text to be classified (i.e., its number of words).

$$T2S2(T, \Phi_j) = \frac{\sum\limits_{v} \mu_{\Phi_j}(v) : v \in (T \cap S_{\Phi_j})}{\sum\limits_{i=0}^{j} \mu_{\Phi_j}(w_i)} \tag{3}$$

In (3), Φ_j is the #hashtagged topic *j* fingerprint, *T* is the set of distinct words in the preprocessed tweet text, $S_{\Phi j}$ is the set of words of the #hashtag *j* fingerprint, and $\mu_{\Phi}(v)$ is the membership degree of word *v* in the #hashtag *j* fingerprint. Essentially, T2S2 sums the membership value of every word *v* that is common between the tweet and #hashtag *j* fingerprint, and divides it by the sum of the top *j* membership values in μΦj(wi) where w ∈ (Φ).

T2S2 tends to 1 when most to all features of the tweet belong to the top words of the fingerprint, and approaches 0 when there are no common words between the tweet and the fingerprint, or the few common words are in the bottom of the fingerprint.

Tweets that have a T2S2 score to a given #hashtagged topic above a given threshold, are considered as being relevant to the topic and are retrieved from the database.

In the MISNIS platform we use the following parameters when performing topic detection: fingerprint size *k*=20, words with less than 3 characters removed during preprocessing (no stopwords are removed and no stemming is performed). Tweets with a T2S2 score above 0.10 are retrieved.

4.2 Fuzzy Uke Word Similarity (FUWS)

Obtaining a successful topic fingerprint depends obviously on the data used to create it. The most usual approach to obtain data to create a topic fingerprint consists in using tweets that are undoubtedly associated with the intended topic, i.e., those that contain an #hashtag created to discuss that topic. However, since there are no rules governing #hashtag creation or #hashtag use in Twitter, anyone can create and use any #hashtag referring to a given topic. It is common that many #hashtags are used to refer the same topic (even if some #hashtags naturally end up being more popular than others). It is therefore of interest to have a method that can help finding similar #hashtags in the database in order to extend the dataset of tweets used to create the fuzzy fingerprint.

Another characteristic of Twitter, is the increasingly higher use of mobile devices such as smartphones. The habit of tweeting using virtual keyboards is usually associated with a relevant number of “thumbos”, i.e., typing errors (aka “typos”) due to thumbs hitting the wrong virtual key due to the small (virtual) keys and lack of physical key feedback. Not even the use of automatic word correction prevents a high number of word errors in tweets’ text. As a result, many tweets relevant to a given topic might not be retrieved when looking for specific keywords because the keywords might contain errors. This is even more likely in the case of #hashtags, since word correction usually does not affect them. Therefore, it is also of interest to detect such errors when creating the dataset of tweets used to create the fuzzy fingerprint.

Detecting similar #hashtags and/or word errors can be accomplished using string similarity techniques. Current research on string similarity offers a panoply of measures that can be used in this context, such as the ones based on edit distances (Levenshtein 1966) or on the length of the longest common subsequence of the strings. However, most of the existing measures have their own drawbacks. For instance, some do not take into consideration linguistically driven misspellings, others the phonetics of the string or the mistakes resulting from the input device. In fact, edit distances, even if universally used for online word typing correction are not adequate at all for the kind of problem we are approaching, since they do not have a strong discriminative power.

In MISNIS we used the Fuzzy Uke Word Similarity (FUWS) (Carvalho and Coheur, 2013) to detect #hashtag/keyword similarity. This word similarity function combines the most interesting characteristics of the two main philosophies in word and string matching (edit distance and common subsequence of strings), and by integrating specific fuzzy based expert knowledge concerning typographical errors (like for example taking into consideration which keys are closer to others and are more likely to appear in “thumbos”), can achieve a good discrimination.

The similarity threshold value used to process word and #hashtag similarity in MISNIS is 0.67, as suggested by Carvalho and Coheur (2013).

4.3 User Influence

In order to assert user influence for a given Twitter topic (using the tweets obtained using the Fuzzy fingerprints), and considering the definitions of influence discussed in Section 2.3, we proposed and implemented a graph representation of user's influence based on mentions (Rosa *et. al*, 2015).

In the proposed user influence representation, whenever a user is mentioned in a tweet's text using the @user tag, a link is made from the creator of the tweet, to the mentioned user:

- When @userA tweets “Do you think we can we get out of this financial crisis, @userB?”, the link @userA->@userB is created.

This is also true for re-tweets:

- The tweet “RT @userC The crisis is everywhere!” from @userA, creates the link: @userA->@userC.

In (Rosa *et. al*, 2015), an empirical analysis showed that, in the context of Twitter User Influence, PageRank (Page 1998, 1999) outperforms other well-known network centrality algorithms, in particular, Katz (1953). As such, PageRank was chosen for the implementation of determining user relevance within MISNIS.

PageRank parameterization consists in deciding the damping factor *d* (see section 2.3). The true value that Google uses as damping factor is unknown, but it has become common to use *d*=0.85 in the literature. A lower value of the damping factor implies that the graph's structure is less respected, therefore making the "walker" more random and less strict. After several experiments we opted to use *d*=0.85 within the platform.

4.4 Sentiment Analysis

Our approach to Sentiment Analysis follows closely the idea explored by (Go, 2009), that consists of using emoticons, abundantly available on tweets, to automatically label the data and then use such data to train machine learning models. In order to build our sentiment models, we have used the knowledge flow described in http://markahall.blogspot.co.nz/2012/03/sentiment-analysis-with-weka.html (retrieved in September, 2015) that uses weka [Hall *et al*., 2009] to automatically label the training tweets based on emoticons.

We have then adopted an approach based on logistic regression, which corresponds to the maximum entropy (ME) classification for independent events (Berger, 1996), to create our sentiment models, based on the previously labeled data. The ME models used in this study were trained using the `MegaM` tool (Daume, 2004), which uses an efficient implementation of conjugate gradient (for binary problems). Finally, the MegaM models were used thought an interface available from the NLTK toolkit[9].

Our models, trained using about 200k tweets, achieve an accuracy between 69% and 71% for the binary classification problem: Positive vs. Negative.

5 Application example

Here we present an execution of the MISNIS framework, applied to a real case collected from the Portuguese database of tweets in MongoDB.

On the night from 21st to the 22nd of November of 2014, former Portuguese Prime Minister José Sócrates was arrested under suspicion of fraud and money laundering during his time in office. This piece of news was the headline of all media outlets for several days and, to this day, remains quite an important story to follow. Naturally, Twitter was no exception, which made this an interesting story to analyze.

The following data was inputted into the interface webforms:

- Start Date: 20th November 2014 00:00:00
- End Date: 23th November 2014 23:59:59
- Keywords: socrates, freesocas

[9] http://www.nltk.org/_modules/nltk/classify/megam.html

With this input, the framework generated the screen shown in Figure **7**.

Out of a total 3309468 tweets in that time interval, 380 were identified to the inputted or other similar keywords (including the hashtags #socrates and #freesocas). These 380 tweets were used to train the Twitter Topic Fuzzy Fingerprint method along with 15 other top trends existing within the full three million tweets. A total of 29019 tweets containing the 15+1 trends were used as the training set.

As an output, the procedure created a dataset of 10687 tweets regarding José Socrates' arrest, produced several indicators regarding topic evolution through time, and listed the top 20 users in discussing and forwarding the topic (Table 3).

The first retrieved true positive tweet occurs on November 22nd, at 00:08:51, just a few minutes after the arrest occurred, which shows once again the relevance of Twitter incurrent news events.

The Precision of the retrieved results is 0.96, and most of the false positives (i.e., tweets that are not related to José Sócrates' arrest) were retrieved on the 48 hours before the arrest (November 20-21st). It was not possible to calculate the Recall since it would be necessary to manually check 3 Million tweets one by one to count the false negatives (this would obviously defeat the purpose of the expert system), but previous tests using the same parameters on other databases have shown that Fuzzy Fingerprint Recall is usually slightly higher than Precision.

It should be noted that, at the time, the database did not contain tweets from Twitter users that never produced geolocated contents. As such, there were no tweets collected from the major media players such as TV stations and newspapers.

Although 10687 tweets out of a universe of 3309468 tweets, seems a low number, this can be explained by the absence in the database of the major media players, and also by the fact that the majority of Twitter users in Portugal are young teenagers (Brogueira, 2014a, 2014b) that usual do not show any special interest in political matters.

Of special relevance is the fact that from 380 hashtagged tweets, the framework retrieved more than 10K tweets that could otherwise go unnoticed among more than 3 million tweets (with a very high Precision).

Amongst the top users it is rather interesting the high incidence of Portuguese comedians, namely @niltoncomedy (ranked 5th), @raminhoseffect (ranked 11th), and @omalestafeito (ranked 13th), and also a few humor oriented internet personalities, such as @avo_idalina (ranked 4th), who is a fictional grandmother character, @miguelluzp (ranked 15th) and @oconguito (ranked 19th), both famous YouTuber teenagers.

A proper sociological analysis on the retrieved results is available in (Rebelo *et. al*, 2016).

Table 3: Top-20 most influential users on Twitter during the early days of the Sócrates prison arrest events (excluding major media players).

Ranking (Page Rank)	User	PageRank Weight
1	@afonsotorrao	0.00701
2	@likerabos	0.00418
3	@sergiohatesyou	0.00356
4	@avo_idalina	0.00295
5	@niltoncomedy	0.00291
6	@tiagosamuel69	0.00219
7	@marinhskilla	0.00209
8	@pedromvfranco	0.00206
9	@nandinholuz	0.00167
10	@unkndeath2002	0.00158
11	@raminhoseffect	0.00155
12	@im_a_barbieee	0.00143
13	@omalestafeito	0.00141
14	@issodepende	0.00134
15	@miguelluzp	0.00127
16	@inaaraujo99	0.00126
17	@rubencgomes	0.00125
18	@pedroboucherie	0.00124
19	@oconguito	0.00109
20	@daspthebest	0.00109

6 Conclusions

Social networks and social networking are here to stay. This is not a controversial or novel statement: independently from how one feels towards adopting the use of social networks, no one can deny their importance in current modern world society. From event advertising or idea dissemination, to commenting and analysis, social networks have become the *de facto* means for individual opinion making and, consequently, one of the main shapers of an individual's perception of society and the world that surrounds her/him.

Despite the undeniable importance of social networks, too many questions concerning their effect in society are yet to be properly addressed. What makes events become important in social networks? Why and how they become important? How long does it take for an event to make an impact in social networks and society? Can social networks give more importance to an event than it really deserves, i.e., are social networks becoming a factor by themselves? What is the role of social networks' major actors (important journalists, bloggers, commentators, politicians, etc.) in the propagation of such events? Are such actors in the origin of the events or mere catalysts to the observations of minor role players?

In this article we present a framework, MISNIS, that can help answering such questions. The framework enables the identification and traction of important events (topics) and of key actors within those topics, as well as identifies their origin and propagation timeline. Measures and indicators to characterize events are provided by the framework contributing with essential information to understand the social network phenomena and its importance in current world society.

MISNIS adresses the issues of collecting, storing, managing, mining and visualizing Twitter data. It applies several well-known and other novel techniques in the fields of Computational Intelligence, Information Retrieval, Big Data, Topic Detection, User Influence and Sentiment Analysis; applied to social networks Data Mining, in particular, Twitter.

MISNIS can be used as an expert system by social scientists, sociologists and other users without (computer) technical (computer) expertise in order to study social networks' impact in society.

The framework is currently fully operational, even if on a prototype form with limited access. External access to its functionalities can be made available by request to the authors.

The framework as been developed with Twitter in mind, but any other public social networks, such as public blogs, webpages, public Facebook profiles, etc., can and will be added as sources of data in a near future.

Acknowledgment

This work was supported by national funds through FCT Fundação para a Ciência e a Tecnologia, under project PTDC/IVC-ESCT/4919/2012 and project PEstOE/EEI/LA0021/2013.

References

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Proc. of LREC'10, Valletta, Malta. ELRA.

Berger, A. L., Pietra, S. D., and Pietra, V. D. (1996). A maximum entropy approach to natural language processing. Computational Linguistics, 22(1):39–71.

Brogueira, G., Batista, F., Carvalho, J. P., and Moniz, H. (2014a). Expanding a database of portuguese tweets. In SLATE'14 3rd Symposium on Languages, Applications and Technologies, volume 4569 of OpenAccess Series in Informatics (OASIcs), pages 275–282. Schloss Dagstuhl.

Brogueira, G., Batista, F., Carvalho, J. P., and Moniz, H. (2014b). Portuguese geolocated tweets: an overview. In ISDOC2014 - Proceedings of the International Conference on Information Systems and Design of Communication, pages 178–179. ACM.

Brogueira, G., Batista, F., and Carvalho, J. P. (2015a). Arquitetura e desenvolvimento de um repositório de tweets em português europeu. In 5as Jornadas de Informtica da Universidade de vora - JIUE 2015. Springer.

Brogueira, G., Batista, F., and Carvalho, J. P. (2015b). Sistema inteligente de recolha, armazenamento e visualização de informação proveniente do twitter. In Conferência da Associação Portuguesa de Sistemas de Informação, CAPSI 2015.

Brogueira, G., Batista, F., and Carvalho, J. P. (2015c). Using geolocated tweets for characterization of portuguese administrative regions. In 18th AGILE International Conference on Geographic Information Science.

Carvalho, J. P. and Coheur, L. (2013). Introducing UWS - A fuzzy based word similarity function with good discrimination capability: Preliminary results. In FUZZ-IEEE, pages 1–8.

Carvalho, J. P., Pedro, V., and Batista, F. (2013). Towards intelligent mining of public social networks' influence in society. In IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pages 478 – 483, Edmonton, Canada.

Cataldi, M., Di Caro, L., and Schifanella, C. (2010). Emerging topic detection on twitter based on temporal and social terms evaluation. In Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD '10, pages 4:1–4:10, New York, NY, USA. ACM.

Cha, M., Haddai, H., Benevenuto, F., and Gummadi, K. P. (2010). Measuring User Influence in Twitter: The Million Follower Fallacy. International AAAI Conference on Weblogs and Social Media, pages 10–17.

Culotta, A. (2010). Towards detecting influenza epidemics by analyzing twitter messages. In Proceedings of the First Workshop on Social Media Analytics, SOMA '10, pages 115–122, New York, NY, USA. ACM.

Das, S. and Chen, M. (2001). Yahoo! for Amazon: Extracting market sentiment from stock message boards. In Proceedings of the Asia Pacific Finance Association Annual Conference (APFA).

Daume III, H. (2004). Notes on CG and LM-BFGS optimization of logistic regression. http://hal3.name/megam/.

Dehkharghani, R., Mercan, H., Javeed, A., and Saygin, Y. (2014). Sentimental causal rule discovery from twitter. Expert Systems with Applications, 41(10):4950 – 4958.

Domingos, P. and Richardson, M. (2001). Mining the network value of customers. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, pages 57–66, New York, NY, USA. ACM.

Feldman, R. and Sanger, J. (2006). Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, New York, NY, USA.

Gerber, M. S. (2014). Predicting crime using twitter and kernel density estimation. Decision Support Systems, 61(0):115 – 125.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford University.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. SIGKDD Explor. Newsl., 11(1):10–18.

Homem, N. and Carvalho, J. P. (2010). Finding top-k elements in data streams. Inf. Sci., 180(24):4958–4974, Elsevier.

Homem, N. and Carvalho, J. P. (2011a). Authorship identification and author fuzzy fingerprints. In 30th Annual Conference of the North American Fuzzy Information Processing Society, NAFIPS2011.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In Proc. of the 10th ACM SIGKDD int. conf. on Knowledge discovery and data mining, KDD ’04, pages 168–177. ACM.

Kasiviswanathan, S. P., Melville, P., Banerjee, A., and Sindhwani, V. (2011). Emerging topic detection using dictionary learning. In Proceedings of the 20th ACM International Con- ference on Information and Knowledge Management, CIKM ’11, pages 745–754, New York, NY, USA. ACM.

Katz, L. (1953). A new status index derived from sociometric analysis. Psychometrika, 18(1):39–43.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In Proc. of COLING ’04. ACL.

Kontopoulos, E., Berberidis, C., Dergiades, T., Bassiliades, N., (2013). Ontology-based sentiment analysis of twitter posts. Expert systems with Applications, 40(10):4065–4074, Elsevier.

Kumar, S., Morstatter, F., and Liu, H. (2013a). Twitter Data Analytics. Springer, New York, NY, USA.

Kumar, S., Morstatter, F., Zafarani, R., and Liu, H. (2013b). Whom should I follow?: Identifying relevant users during crises. In Proceedings of the 24th ACM Conference on Hypertext and Social Media, HT ’13, pages 139–147, New York, NY, USA. ACM.

Lachlan, K. A., Spence, P. R., and Lin, X. (2014). Expressions of risk awareness and concern through twitter: On the utility of using the medium as an indication of audience needs. Computers in Human Behavior, 35(0):554–559.

Leavitt, A., Burchard, E., Fisher, D., and Gilbert, S. (2009). The influentials: New approaches for analyzing influence on twitter.

Lee, K., Palsetia, D., Narayanan, R., Patwary, M. M. A., Agrawal, A., and Choudhary, A. (2011). Twitter trending topic classification. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, ICDMW ’11, pages 251–258, Washington, DC, USA. IEEE Computer Society.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals” Soviet Physics Doklady, 1966. 10:707–710.

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., and Miller, R. C. (2011a). Tweets as data: Demonstration of tweeql and twitinfo. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, SIGMOD ’11, pages 1259–1262, New York, NY, USA. ACM.

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., and Miller, R. C. (2011b). Twitinfo: Aggregating and visualizing microblogs for event exploration. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’11, pages 227–236, New York, NY, USA. ACM.

Mathioudakis, M. and Koudas, N. (2010). Twittermonitor: Trend de- tection over the twitter stream. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD ’10, pages 1155–1158, New York, NY, USA. ACM.

Mazzia, A. and Juett, J. (2010). Suggesting hashtags on twitter. Master’s thesis, University of Michigan.

Mendoza, M., Poblete, B., and Castillo, C. (2010). Twitter under crisis: Can we trust what we rt? In Proceedings of the First Workshop on Social Media Analytics, SOMA ’10, pages 71–79, New York, NY, USA. ACM.

Oussalah, M., Bhat, F., Challis, K., and Schnier, T. (2013). A software architecture for twitter collection, search and geolocation services. Knowledge-Based Systems, 37(0):105 – 120.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). 1 Introduction and Motivation 2 A Ranking for Every Page on the Web. World Wide Web Internet And Web Information Systems, 54(1999-66):1–17.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of EMNLP, pages 79–86.

Paul, M. J. and Dredze, M. (2011). You are what you tweet: Analyzing twitter for public health. In Adamic, L. A., Baeza-Yates, R. A., and Counts, S., editors, ICWSM. The AAAI Press.

Perera, R., Anand, S., Subbalakshmi, K., and Chandramouli, R. (2010). Twitter analytics: Architecture, tools and analysis. In MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010, pages 2186–2191.

Phuoc, N. Q., Kim, S.-R., Lee, H.-K., and Kim, H. (2009). Pagerank vs. Katz status index, a theoretical approach. In Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, ICCIT ’09, pages 1276–1279, Washington, DC, USA. IEEE Computer Society.

Qu, Y., Huang, C., Zhang, P., and Zhang, J. (2011). Microblogging after a major disaster in china: A case study of the 2010 yushu earthquake. In Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work, CSCW ’11, pages 25–34, New York, NY, USA. ACM.

Razis, G. and Anagnostopoulos, I. (2014). Influencetracker: Rating the impact of a twitter account. CoRR.

C. Rebelo, I. Pereira, H. Rosa, F. Batista, J.P. Carvalho, “The news will be tweeted: multiple uses of Twitter around a major political event”, submitted to New Media and Society.

Rogers, E. M. (1962). Diffusion of innovations.

Rosa, H., Carvalho, J. P., Astudillo, R., and Batista, F. (2015). Detecting user influence in twitter: Pagerank vs katz, a case study. 7th European Symposium on Computational Intelligence and Mathematics.

Rosa, H. (2014). Topic detection within social networks. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa.

Rosa, H., Batista, F., and Carvalho, J. P. (2014a). Twitter topic fuzzy fingerprints. In WCCI2014, FUZZ-IEEE, 2014 IEEE World Congress on Computational Intelligence, International Conference on Fuzzy Systems, IEEE Xplorer, pages 776–783, Beijing, China.

Rosa, H., Carvalho, J. P., and Batista, F. (2014b). Detecting a Tweet's Topic within a Large Number of Portuguese Twitter Trends. In Pereira, M. J. V., Leal, J. P., and Simes, A., editors, 3rd Symposium on Languages, Applications and Technologies, volume 38 of OpenAccess Series in Informatics (OASIcs), pages 185–199, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Saha, A. and Sindhwani, V. (2012). Learning evolving and emerging topics in social media: A dynamic nmf approach with temporal regularization. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12, pages 693–702, New York, NY, USA. ACM.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: Real-time event detection by social sensors. In Proceedings of the 19th International Conference on World Wide Web, WWW '10, pages 851–860, New York, NY, USA. ACM.

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., and Sperling, J. (2009). Twitterstand: News in tweets. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09, pages 42–51, New York, NY, USA. ACM.

Santos, C. J. and Matos, S. (2013). Predicting flu incidence from portuguese tweets. In International Work-Conference on Bioinformatics and Biomedical Engineering 2013. Proceedings.

Scanfeld, D., Scanfeld, V., and Larson, E. L. (2010a). Dissemination of health information through social networks: Twitter and antibiotics. American Journal of Infection Control, 38(3):182–188.

Stone, P., Dunphy, D., Smith, M., and Ogilvie, D. (1966). The General Inquirer: A Computer Approach to Content Analysis. MIT Press.

Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proc. of the 40th Annual Meeting on Association for Computational Linguistics, pages 417–424. ACL.

Twitter (2010). To trend or not to trend. https://blog.twitter.com/2010/trend-or-not-trend. Accessed: 2014-03-28.

Widener, M. J. and Li, W. (2014). Using geolocated twitter data to monitor the prevalence of healthy and unhealthy food references across the {US}. Applied Geography, 54(0):189 – 197.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In Proc. of HLT/EMNLP, pages 347–354. ACL.