ISBN: 978-989-8533-41-8 © 2015

A SERVICES FRAMEWORK FOR USING THE SOCIAL WEB AS SENSOR WEB: IDENTIFYING EVENTS IN REAL TIME USING SPATIAL, TEMPORAL AND SEMANTIC REFERENCES (SUSS)

Dimitris Kotzinos[1], Sofia Kleisarchaki[2], Nicolas Spyratos[3], Lila Theodoridou[4], Nikoloas Petalidis[4] and Panagiotis Kazakis[4]
[1]*ETIS/UCP, 2 av. Adolphe Chauvin, 95000 Pontoise, France*
[2]*University of Crete, Heraklio, Greece*
[3]*LRI, Paris, France*
[4]*TEI of Central Macedonia, Serres, Greece*

ABSTRACT

In this paper we discuss the design of a system that uses the social networks to detect events using the social networks' users as sensors that report information. The system is using text mining and machine learning algorithms to cluster tweets and social media posts and then use spatial and temporal metadata to identify events, as well semantic information to identify the sematic relations among the different concepts involved. We describe the steps to follow and the algorithms we implement in the system in order to achieve this. At the end we also discuss the overall user functionality offered to users interested in different events in diverse contexts.

KEYWORDS

Event detection, Social web, social sensor web, tweets clustering

1. INTRODUCTION

This papers aims at describing the design of a services framework for using the social web as sensor web, with the up most target to identify events in real time**.** We describe what the system should be able to do and to some extend how we plan to do this; at least the part where the knowledge of what is feasible blends with the requirements imposed by the users of the system. Moreover we want to describe the system requirements; namely we try to describe what the system requires to function in terms of data, user input and relevant parameters. It should be noted here that the aim of this paper is to build a prototypical system and not a full-blown commercial system that can sustain many users and deal with all kinds of user requirements. Moreover we tried to identify useful capabilities of such a system in order to advance our research and provide a solid basis for this research to expand in the near future.

The document is structured as follows: after this short introduction, the architecture of such a service based system is described in section 2, which details the application layer of the system (where most of the computations take place) and the data layer of the system where the necessary data transformations happen and the data are stored; we also describe briefly the presentation layer. The document concludes in section 3 by presenting the highlights of the research so far and introducing the future directions.

2. ARCHITECTURAL DESIGN OF SUSS

In this paper we present the high-level functionality of SUSS along with the architecture of the prototype system. The proposed architecture broadly distinguishes three generic conceptual and also development layers that best capture and express the functionality of the system. The first upper level concerns the

presentation layer that is a vitally important aspect of the system and a critical part to its success. After all, the presentation layer represents the interface between the user and the rest of the application and thus we aim to provide users with an easily interactive, efficient and effective front end. The implementation of this layer's functionalities concludes to an online real-time GUI-based system that will notify users for new events of their interest that are being discovered and detected as new posts arrive over time.

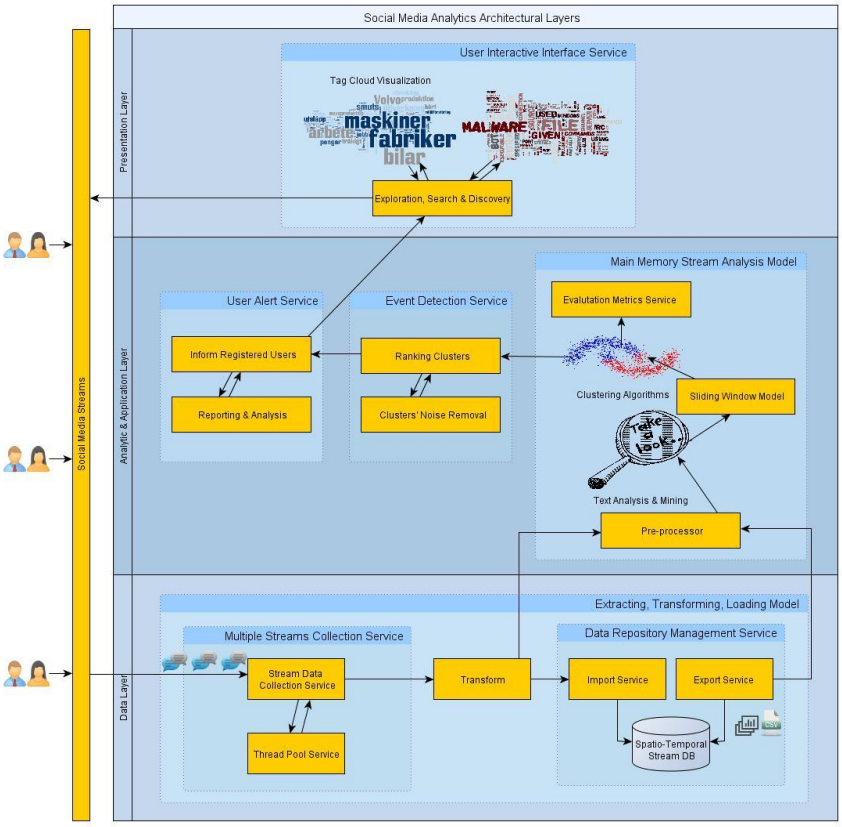

Figure 1. The layered architecture of SUSS

The second layer of *application and social analytics* encompasses the major functionality that allows the collection, storage from one or more social networking sites, analysis and extraction of specific events as they are occurring. In particular, it involves fast algorithms and robust data structures that are designed to operate in real-time the processing of large-scale data with high arrival rates and volumes. This layer provides a services API that can be used by programmers in order to build new applications on top. In summary, this comprehensive and extensible programmable layer would allow event detection in social networks going beyond the traditional text based approach by exploiting the social, temporal, spatial and of course textual dimensions of the social networks. The lower *data layer* consists of an Extracting, Transforming and Loading (ETL) model that is responsible to pull social data from online media and transform them into an acceptable database format. The loading service is responsible to store and retrieve data of spatio-temporal characteristics and feed them in an online manner to the main memory model. The design and maintenance of this layer appears several challenging issues especially when dealing with heterogeneous social media sources.

All of the three layers are further explained in the subsequent sections in a bottom-up series. Furthermore, an overview of the architectural design of SUSS is depicted in Figure 1. The functionality of the various logical components, partially described earlier in this section, are illustrated by a series of interrelated services. The services corresponding to each one of these modules are described along with the proposed functionality for each one.

ISBN: 978-989-8533-41-8 © 2015

2.1 Presentation Layer

A main objective of this component is to provide a graphical user interface (GUI) of bundled web services that would be accessible by multiple concurrent users. This GUI is able to present processed and combined information from various social media streams concerning real-time news and events and updates this information as new posts become available. The main concept of this task is to provide users with a friendly, concise, and unambiguous framework for online real-time information awareness.

The users of this framework are alerted for new detected events and are able to search and discover discussions as they occur in the social web. An updatable list of events is presented to the users' profile page ranked in a chronological order and filtered by the system for being event-related and spam-free. The service covers events appearing in a specified period of time (e.g., last few days) and removes older posts guaranteeing compact and fresh content. Each presented event is a summary of the detected relative posts accompanied with external links to other sources and representative users' posts of high popularity. Moreover, the users are not only allowed to navigate through different events but they can explore different aspects and opinions inside an evolving topic and being informed about the diversity of users' point of view. Thus, related stories corresponding to different aspects of an event are grouped together and presented to the end users in a graph-based form, similar in some way to the Google News[1] presentation policy.

Figure 2. Tag Clouds for (a) Libya Revolution (b) Earthquake in Japan

Moreover, several tools for creating graphical representations of the social content (e.g., cluster's content or events' summary) built upon the overall system are provided. The programmers are able to obtain an easy and synoptic way to illustrate the content of a detected event from social streams. To this end, we provide functionality through the public API for tag clouds. A tag cloud or word cloud is a visual representation for text data, typically used to depict the textual content of a cluster. Tags are usually single words and give a quick notion of the cluster's thematic area and discussion topic. The importance of each tag is shown with font size or color. The word with the highest size or with the most intense color depicts a high importance (e.g., maximum frequency occurrence) inside the cluster. An example of tag cloud is shown in Figure 2 where the contents of two distinct clusters are illustrated. The first Figure 2a refers to the Libya revolution against the regime, where the words like 'Libya', 'gaddafi' and 'war' dominating inside the cluster. The second Figure 2b summarizes the earthquake and tsunami that occurred in Japan and shocked the local society. The prevailing words 'Japan', 'earthquake', 'tsunami', 'nuclear' give a quick notion of the sudden disaster.

Thus, this visual summary of the clusters is useful for quickly perceiving the most prominent terms and enriches the knowledge of the keywords for a topic. On the other hand, it can also be used to summarize the events occurring in a period of time simultaneously giving an intuition of each event's popularity. Both ways of exploiting the technique are available through the API. The implementation of this module allows users to easily explore, search and discover new interesting topics in a compact and elegant way.

2.2 Analytic & Application Layer

2.2.1 Main Memory Stream Analysis Model

The major functionality of this layer is provided by the main memory stream analysis component, which consists of five simpler modules. This core component is either associated with the stream collector service pulling almost real-time social content from the streams or with the exporter service from the lower level, which retrieves data from the database in batches or in an incremental mode and feeds them to the operations of the subsequent components.

[1] https://news.google.com

The first sub-component of pre-processing supports a complete tool for extracting and cleansing textual data. The pre-processing procedure of lexical analysis has a great impact on the quality of the final results and consists of five main steps where the input plain text is converted into tokens for the analysis through natural language processing (NLP) methods. The Figure 3 illustrates the steps, which are explained below. Our analysis is built upon English textual content and thus a system for language detection is provided. The prevailing and efficient technique for extracting content oriented to the language is by utilizing dictionaries and maintaining only the social posts with high percent of common words with the given vocabulary. This process can be applied in a streaming mode or offline for experimental purposes.

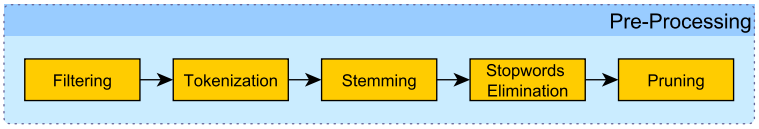

Figure 3. Five Steps of pre-processing component

After the removal of non-English posts, the first step of the pre-process is *filtering* which is applied under specified patterns removing unnecessary tags for the analysis (e.g. HTML or XML) or network-specific keywords. For instance, twitter posts usually contain tags for annotating the replies tweets (e.g., @username) or keywords like 'RT' for mentioning the re-tweets as well as abbreviated URLs. The provided API allows the optional removal of those non-beneficial for the analysis tags that are oriented to the specified social media idioms. Then, the *tokenization* step splits the sentences into tokens of words either by white spaces split criteria or in some cases with more sophisticated techniques, such as grammatical structure extraction. The implementation of this sub-component is focused on techniques that are more suitable for the social content. Thus, the grammatical and syntactical errors existing in social posts do not bode well for implementing tokens' extractor based on their grammatical or syntactical structure. The extracted tokens are converted to their stems, base or root form, through the *stemming* process. This process is crucial for the final results, as it reduces inflected (or sometimes derived) words to their stem. Although the produced stem might not in itself be a valid root, it is usually sufficient that related words map to the same stem. This stem might also not be identical to the morphological root of the word. Several algorithms is implemented for this task, with the most well known among them the Lovins stemmer [8] and Porter's stemmer [12] for the English language. Finally, in order to reduce memory and time complexity a *stopword elimination* and *pruning* step is applied. In the former, a list of words that are very commonly used in English language and frequently met in texts are provided in order to drop these words at indexing time before the analysis. In the latter, a basic pruning functionality is supported to remove words of low frequency through the corpus by defining a lower frequency threshold.

The products of the pre-process step are given as input to the text mining and analysis procedure in order to create the final vector space that is exploited by the clustering algorithms. In particular, this sub-component contains implemented functionality to support feature extraction techniques. It supplies analysts with tools able to transform the high-dimensional space of data into a space of fewer dimensions by performing linear mapping of data producing a low maximum relative error. A well-known unsupervised method that is provided for this purpose is the Principal Component Analysis (PCA) [13], which is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components[2]. Furthermore, an additional helpful tool gives the basic functionality of Named Entity Recognition (NER) that determines which items in the text map to proper names, such as people or places and what the type of each such name is (e.g. person, location, organization). The highlight of informative entities like Named Entities as well as enrich the textual content (for example, the vocabulary of a tweet can be extended using Yahoo!Boss) would benefit the data management. Finally, several weighting schemes are available to deal with the peculiarities of social streams such as their high heterogeneity in discussions and their emerging nature. Among them are the implementations of the well-known weighting schemes TF-IDF[3] and okapi BM25[4] as well as the [7, 14] method. The latter is an incremental weighting method with a dynamic view of

[2] https://en.wikipedia.org/wiki/Principal_component_analysis
[3] http://en.wikipedia.org/wiki/Tf-idf
[4] http://en.wikipedia.org/wiki/Okapi_BM25

the time dimension. The third sub-component includes the implementation of various window model techniques for processing data in smaller batches. The usage of sliding window model gives the ability to the clustering algorithms to remember a past period of time and tune the capacity of this memory. The knowledge of this short past can benefit the clustering quality. Both sliding window models are given to programmers through the API. Finally, the next important step includes the implementation of several clustering methods. So far, we realize the necessity of utilizing different type of algorithms based on the various types of social data in time oriented to the particular analysis problem. Thus, the implementation and sequentially use of an algorithm has value if it belongs to the solution space of the social analytics. This space is located in four dimensions with its pillars shown in Figure 4. The fourth dimension of time is implied and implicitly affects each one of the three axes of the analysis space.

So far, many clustering algorithms have been proposed in literature especially for the task of topic detection in documents belonging in one of three general categories of the x-axis. The first category of partitioning algorithms divide the initial dataset into sets of k clusters, where each object belongs to only one cluster. They create a one-level, un-nested partitioning of the data points where the number of clusters is given as input. We intend to offer a programming API for several well-known algorithms of this type, such as k-means, DBScan and denStream [4, 6, 14]. However, these algorithms suffer from hard parametrization depending on the characteristics of the processing social data. Thus, we provide several techniques for proper initialization of these parameters through a training phase that is based on exploring subset of the corpus utilizing unsupervised evaluation metrics. In the second category, we are dealing with hierarchical methods that produce a hierarchy of clusters, where each cluster is nested into another. The hierarchy, usually presented in a dendogram, can be considered as bottom-up (i.e., agglomerative) or top-down (i.e., divisive). To this end, the TStream [5, 14] algorithm appears interesting properties as it detects spherical clusters of high-similarity (i.e.sub-topics) nested within wider ones (i.e. topics) promising to capture the homogeneity (e.g., different aspect of a discussion) of a topic. The last category of probabilistic algorithms expresses a point's membership in the cluster by probabilities. We are confident that the Latent Dirichlet Allocation method[5] (LDA) and the Locality Sensitivity Hashing (LSH) applied in [11] will have great scientific value in social analytics and thus we provide an implementation for them.

As it is already highlighted, the selection of the proper machine learning algorithm depends on the properties of the social media stream as well as on the focus of the given analysis problem. An enlightening observation over this statement is, for instance, the trend detection problem that is inseparably connected with the emerging tendency of the incoming data. Recent works of trends detection [3, 9] aim to recognize topics of conversations for which there is a sudden rise of interest among people. For example, Twitter shows on its main page few topics, represented by a short number of words or hashtags, with a bursty behavior among tweets. These tweets either mention breaking news (e.g.,an earthquake), government actions (e.g., for a tax reformation low) or any other activity(e.g., comment of an artist, etc.) that become high popular among people. However, these techniques fail to detect the heterogeneity of the social network's discussions, as many topics of high popularity do not necessarily exhibit a sudden emerging behavior. On the contrary there exist periodically rising topics of that behavior with no-news content (e.g., #followfriday) that are incorrectly being detected as interesting and new from these methods.

Given that the majority of Twitter users post messages regarding their personal concerns and matters [10], topic detection and more particularly event detection attempts to recognize real-world events that could be of more general interest and disambiguate their spatio-temporal context. To this end, multi-features similarity metrics that combine textual content and the community's structure along with the spatio-temporal characteristics of social posts have been designed and provided through the public API. Recent works usually ignore the four dimensions of the problem appearing weakness to realize the importance of defining the requirements and the peculiarities of the temporal social content in their problem analysis context. Thus, there is a lack of solution space confinement and unconsciously selection of algorithm for solving the analysis problem for the given dataset. For this purpose, we would like not only to provide the implementation of several clustering algorithms from all of the three categories as well as the potential to build and test on representative samples but also the knowledge for conscious selection of a clustering algorithm. Last but not least, a variety of supervised and unsupervised quality metrics are available for the evaluation of the algorithms. The set of supervised metrics includes Purity, Precision, Recall, F-Measure, conditional Entropy and Normalized Mutual Information (NMI). The set of unsupervised learning supports intra-cluster separation measures, like Between Sum of Squares (BSS) and between cluster cohesion measures, like Within Cluster Sum of Squares Error (WSS) and Silhouette Index.

[5] https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

2.2.2 Event Detection Service

Social media are precious sources of information as they contain news that are not present in the newswire and also they tend to break events sooner than traditional news media. On the other hand, social media contain a more social-oriented content than news propagation mechanisms making this user-generated content suffering from a high level of noise. Especially, in the context of event detection any information that is not related to an event can be considered as noise. In fact, a study in 2009 showed that only 3.6\% of all tweets are news-related, which means that the level of noise reaches a percent of 96.4\%. In order to alleviate this problem we need a methodology to decide whether a new cluster is about an actual event or it has to be discarded.

The main idea to overcome this challenging issue is the use of sliding window models. The execution of clustering process inside sliding windows introduce a latency and therefore a data accumulation that allows the system to decide in a post-processing step if the produced clusters are containing a valuable event. There are several strategies in which clusters can be exploited in order to improve the quality of the final detected events. We design, explain and implement five distinct criteria, which ranks the produced cluster at regular times and present to the final user only the contents of the event-labeled clusters. The first ranking criterion would be to use the size of clusters and maintain only those whose size exceeds a pre-defined threshold. This technique encapsulates the notion of emerging trends and highlights the events of high popularity, by reflecting the idea that clusters of many similar posts indicate a new uprising topic of interest. In the same context, the second criterion is presented where the clusters of the most unique users discussing about the topic are ranked in the top of the list. The authors of [11] have also introduced the metric of entropy as a ranking criterion. They observed that clusters with high percent of spam content have very low entropy. The implementation of this metric as well as the combination of it with the previous criteria is provided. In particular, the clusters of highest entropy with the most unique users will conquer the top positions in the ranking list. In the case of Twitter posts' clustering, a more content-based ranking approach will rank the tweets according to the number of distinct hashtags they contain. An evolving topic discussion of a current event is more probable to generate more related hashtags over time (e.g., #japan, #earthquake, #tsunami etc.) than an every-day discussion (e.g., #ff, #tfb). Another criterion that is also related with the content is the rank of clusters with the highest similarity of their posts to the top. The intuition behind this technique is that posts' homogeneity and thus cluster's cohesion predetermine a probable event occurrence. The criteria described above are implemented in order to test their value in the noise reduction task and are evaluated for their efficiency in producing reports to the users of only pure and sensible events content.

2.2.3 User Alert Service

The free online service of user alert refers to the delivery of user-subscribed information to the user. This service automatically notifies users for new content that satisfies their criteria with notifications and content updates appearing on the user's profile page. In this section we will define the types of alerts that are supported by our system.

There exist four types of alerts sent when new content matches the search criteria: (1)*Everything* - It is the default setting, that aggregates and presents all events detected by the system; (2) *Thematic Criterion* - The user defines criteria based on his/her interests of topics; (3) *Property-based Criterion* - The user defines criteria on non-thematic properties (e.g., geographic region); (4) *User's prefe*rences - No criterion is applied. The system is trained to recognize users' preferences.

The default setting of the system aggregates and presents all the detected events as they happen. In the main page of each user a list of events are shown and updated with no frequency control. That means, that the users have no control of how often they receive alerts of new content and this option sets the maximum frequency of alerts. The order of the events is chronological with the most recent appearing on the top of the list and the thematic area of the topics is unspecified. In particular, there is no ranking among the presented events and different thematic areas appear simultaneously (e.g., political and athletic news).

The second type of alerts gives the user the potentiality to tune the system and define the thematic areas of his/her interest that (s)he wants to be alerted. To this end, several categories are provided and among them are *political news*, *athletic* and *social events* as well as the opportunity to the user to define keywords in order to be alerted when the new content matches the user-selected terms. Furthermore, the user has the choice to define the frequency of checks for new results. Three options are available "once a day", "once a week" or "as it happens" which also is the default choice.

The next alert type supports property-based criteria set by the users. These criteria are independent from the events and concern the meta-data extracted from the social networks. One criterion supported by the system is the geographical specifications given by the user that satisfies queries of the form "I am interested in events from the specified geographical region". In the same sense, a criterion concerning the network's structure is supported. The user is able to define other users of his/her network that (s)he is interested in getting informed each time those friends are contributing in a detected event. The same policy, as previously mentioned, is followed about the frequency of updates' checks.

Last but not least, the system is self-learning and self-training by providing the functionality to the end user, if (s)he wishes, to allow the system to learn his/her preferences over time. By enabling this option the user can provide feedback to the system concerning his/her preferences by rating the given results and explicitly rank them indicating his/her interests. Through this process the system is trained to return only relevant results to the user's taste. The designed alert service is of a great importance in order to hold the users' attention and increase the time spent in the application. A successful alert service can guarantee the success of the system.

2.3 Data Layer

The lower layer of data management has the role of pulling the social content from the social media in multiple ways. For the case of Twitter networking site, the provided API is easily parameterized in order to retrieve data under given keywords or from specified users etc. either through the search API or in a streaming mode[6]. Furthermore, a real-time allocation of streaming data is available without keywords declaration. For the purpose of the multiple streams collection service a pool of threads is implemented to manage the incoming data and store them in the data warehouse. The stream data collector provides a scalable persistence service for retrieving and storage large volumes of social streams data from heterogeneous sources. The communication with the data repository management service is possible through transformation and wrapper interfaces that convert data into proper database format and encapsulate the functionality of the data warehouse providing a level of abstraction.

The data collection includes textual posts but also information of the meta-data accompanying the social messages, such as the author's name and friend list, the creation time, the geo-location of the post etc. An Entity-Relationship diagram is shown in Figure 5 that illustrates the available Twitter information in its entirety and the corresponding type in a relational database schema. Finally, the Exporter component of data repository management service gives the ability for easily retrieving data in varying formats. Several formats are provided to the analysts, like .svg, .csv, .html or Google charts, in order to be able to extract valuable statistics and results. Furthermore, a communication is established with the higher layer of social content analysis in order to feed the analysis algorithms and the clustering techniques either with real-time social stream's data or with archived data oriented to the analysis purpose.

3. CONCLUDING REMARKS

This paper introduces the functional requirements of the system and at the same time tackles these requirements by providing the design choices that specify a service-oriented system. We provided a layered service oriented architecture that satisfies the requirements for such a system while supports diverse clients that can use the system services. The system has been developed as a functional prototype consisting of a set of services published through a comprehensive API.

[6] https://dev.twitter.com/docs/using-search

ACKNOWLEDGEMENT

This research is implemented through the Operational Program "Education and Lifelong Learning" and is co-financed by the European Union (European Social Fund) and Greek national funds.

REFERENCES

1. Twitter api. http://apiwiki.twitter.com/.
2. F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In *2006 SIAM Conference on Data Mining*, pages 328–339, 2006.
3. M. Cataldi, L. Di Caro, and C. Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proc. of the 10th MDMKDD*, pages 4:1–4:10, NY, USA, 2010. ACM.
4. L. Devroye. Sample-based non-uniform random variable generation. In *Proc. of the 18th WSC*, pages 260–265, NY, USA, 1986. ACM.
5. M. Ester, H. Peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. *AAAI Press*, 1996.
6. J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.
7. C.-H. Lee, C.-H. Wu, and T.-F. Chien. Burst: a dynamic term weighting scheme for mining microblogging messages. In *Proc. of the 8th ISNN* - Volume Part III, pages 548–557, Berlin, Heidelberg, 2011. Springer-Verlag.
8. J. B. Lovins. *Development of a Stemming Algorithm*. June 1968.
9. M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proc. of SIGMOD*, pages 1155–1158, NY, USA, 2010. ACM.
10. M. Naaman, J. Boase, and C.-H. Lai. Is it really about me?: message content in social awareness streams. In *Proc. of the 2010 ACM conference on Computer supported cooperative work, CSCW '1*0, pages 189–192, New York, NY, USA, 2010. ACM.
11. S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *HLT: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189, Stroudsburg, PA, USA, 2010.
12. M. F. Porter. Readings in information retrieval. chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
13. J. Yang and Z. Ma. Document clustering based on mutual information and pca subspace. In *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, pages 2983 –2986, aug. 2011.
14. M. Zimmermann, I. Ntoutsi, Z. F. Siddiqui, M. Spiliopoulou, and H.-P. Kriegel. Discovering global and local bursts in a stream of news. In *Proc. of the 27th SAC*, pages 807–812, NY, USA, 2012. ACM.