Sensing trending topics in Twitter

Luca Maria Aiello, Georgios Petkos, Carlos Martin, David Corney, Symeon Papadopoulos, Ryan Skraba, Ayse Goker, Yiannis Kompatsiaris, Alejandro Jaimes

***Abstract*—Online social and news media generate rich and timely information about real-world events of all kinds. However, the huge amount of data available, along with the breadth of the user base, requires a substantial effort of information filtering to successfully drill down to relevant topics and events. Trending topic detection is therefore a fundamental building block to monitor and summarize information originating from social sources. There is a wide variety of methods and variables and they greatly affect the quality of results. We compare six topic detection methods on three Twitter datasets related to major events, which differ in their time scale and topic churn rate. We observe how the nature of the event considered, the volume of activity over time, the sampling procedure and the pre-processing of the data all greatly affect the quality of detected topics, which also depends on the type of detection method used. We find that standard natural language processing techniques can perform well for social streams on very focused topics, but novel techniques designed to mine the temporal distribution of concepts are needed to handle more heterogeneous streams containing multiple stories evolving in parallel. One of the novel topic detection method we propose, based on n-grams, cooccurrence and df-idf_t topic ranking, consistently achieves the best performance across all these conditions, thus being more reliable than other state-of-the art techniques.**

***Index Terms*—Topic detection, Twitter, FA Cup, Super Tuesday, US Elections, Social Sensor**

I. Introduction

The pervasiveness of online social media has seen unprecedented expansion in recent years. As social networking services progressively diffuse in more geographical areas of the world and penetrate increasingly diverse segments of the population, the value of information that is collectively generated on such online platforms increases dramatically. In fact, interactions and communication in social media often reflect real-world events and dynamics; as the user base of social networks gets wider and more active in producing content about real-world events almost in real-time, social media streams become accurate *sensors* of real-world events.

The riots during the Arab Spring [1], [2] the dramatic incidents determined by natural disasters [3] as well as the process of opinion formation around major political themes [4] offer examples of events that have been reported almost in real-time by social network participants. As a result, social media data mining, originally aimed at understanding and predicting the evolution of the online social worlds [5], [6], [7], [8], [9], is now increasingly leveraged to study the dynamics of real-world events. The ability to monitor such phenomena has direct implications on the possibility of understanding and describing real-world events, with applications in the fields of computational journalism [10], [11], urban monitoring [12], and many more. Finally, since social media could be used to manipulate the course of online and offline human dynamics [13], [14], [15] (e.g., by augmenting the consensus on politicians for electoral purposes), detecting anomalous activity can help prevent possible misuses of online social platforms.

To monitor and detect all these aspects in real time, we need to extract the relevant information from the continuous stream of data originating from such online sources. Determining which are the *topics* being discussed by the crowd is the first step towards a high-level, human-understandable description of the social data stream. The task of Topic Detection and Tracking [16] has been tackled in the past for static document corpora, but in a social media context there are many additional factors to consider such as the fragmentation and noise of the user generated content, the strict *real-time* requirement, the *burstiness* of events and their *time resolution*.

We explore how much these factors impact the topic detection results by exploring two orthogonal dimensions: a) the effect that the nature of the input data, including the pre-processing phase, has on the topic detection outcome; and b) the behaviour of different topic detection algorithms themselves.

The methods we test cover three different classes: probabilistic models (Latent Dirichlet Allocation), classical Topic Detection and Tracking (a common document-pivot approach) and feature-pivot methods. Along with this series of methods, we develop four novel approaches, including methods that use the concept of frequent itemset mining.In particular, we show that a method that leverages n-gram cooccurrences (instead of unigrams) and df-idf_t topic ranking is consistently the best performing method among the ones tested. The proposed df-idf_t is a score for burstiness detection that can significantly assist in determining the most rapidly emerging topics. The diversity of the methods presented and the different attributes of the datasets considered (with respect to time-scale and breadth of topical discussions) enable a comparison across several crucial dimensions inherent in the topic detection task that have not been explored in previous work.

The evaluation of methods focuses on a scenario of *sensing*

L. M. Aiello and A. Jaimes are with Yahoo! Research Barcelona ({alucca,ajaimes}@yahoo-inc.com). G. Petkos, S. Papadopoulos, and Y. Kompatsiaris are with the Information Technologies Institute, CERTH, Thessaloniki, Greece ({gpetkos,papadop,ikom}@iti.gr). C. Martin and D. Corney are with Department of Computer Science, City University London, London EC1V 0HB ({martin.carlos.1,david.corney.1}@city.ac.uk). Ryan Skraba is from Alcatel-Lucent Bell Labs, Paris, France (ryan.skraba@alcatel-lucent.com). A. Goker is with Robert Gordon University, School of Computing / IDEAS Research Institute Aberdeen, United Kingdom (a.s.goker@rgu.ac.uk). The majority of this work was conducted while A. Goker was at City University London. This work is supported by the SocialSensor FP7 project, partially funded by the EC under contract number 287975.

real world topics of the kind that would be of interest to the reader of a news portal. We use three large datasets collected from Twitter, for which the sets of ground truth topics have been produced by examining news stories appearing in the mainstream media. The selected datasets the domains of politics (the US Super Tuesday primaries of March 2012 and the US Presidential elections of November 2012) and sports (the English FA Cup Final).

In short, our contributions can be summarized as follows:

- We present a comparative study of a wide range of topic detection methods across three large Twitter datasets on a real-world events sensing scenario. The main idea of using different datasets is to compare the performance of the algorithms in different domains having their own special features.
- We analyze how factors such as the type of input data (e.g. time span, topic breadth) and pre-processing techniques can affect much the quality of topic detection results.
- Among all the methods we test, we find that our novel algorithm combining n-grams with $df\text{-}idf_t$ ranking performs best, and better than other state-of-the-art techniques.

The remainder of the paper is structured as follows. Section II provides an overview of state-of-the-art approaches for topic detection. Section III presents the topic detection methods that are examined in this work, together with some pre-processing steps. Experimental results are presented and discussed in Section IV.

II. Related work

Topic Detection and Tracking (TDT) aims at extracting topics from a stream of textual information sources, or *documents*, and to quantify their "trend" in time [16]. This work focuses on pieces of texts (posts) produced within social media platforms.

Methodologically, general-purpose topic detection can produce two types of complementary outputs: either the documents in the collection are clustered or the most important *terms* or *keywords* are selected and then clustered. In the first method, referred to as *document-pivot*, a topic is represented by a cluster of documents, whereas in the latter, commonly referred to as *feature-pivot*, a cluster of keywords is produced instead.

Both methods have advantages and disadvantages. Document-pivot methods suffer from cluster fragmentation problems and, in a streaming context, they often depend on arbitrary thresholds for the inclusion of a new document to an existing topic. On the other hand, feature-pivot methods are commonly based on the analysis of associations between terms, and often capture misleading term correlations. In general, the two approaches can be considered complementary and, depending on the application, one may be more suitable than the other. In the following, we review several popular approaches that fall in either of the two categories. We also characterize them based on a number of important features, such as incremental computation vs. batch mode or the usage of additional sources of information.

A. Document-pivot methods

Simple document-pivot approaches cluster documents by leveraging some similarity metric between them. The work by Phuvipadawat and Murata [17] follows this direction to provide a method for breaking news detection in Twitter. Tweets retrieved using targeted queries or hashtags are converted into a bag-of-words representation weighted with boosted $tf\text{-}idf$ (term frequency–inverse document frequency) emphasizing important entities such as names of countries or public figures. Tweets are then incrementally merged by considering the textual similarity between incoming tweets and existing clusters. Similar approaches based on textual similarity and $tf\text{-}idf$ can be found in literature [18], [19]. Among them, the method discussed by Becker et al. [19] additionally considers the classification of tweets as referring to real-world events or not. The classifier is trained on a vast variety of features including social aspects (e.g., number of mentions) and other Twitter-specific features. An important drawback of the method is the need for manual annotation of training and test samples.

Dimensions other than text can also be used to improve the quality of clustering. TwitterStand [20] uses a "leader-follower" clustering algorithm that takes into account both textual similarity and temporal proximity. Each cluster center is represented using a centroid $tf\text{-}idf$ vector and the average post-time. A similarity metric based on both dimensions and on the number of shared hashtags allows incremental merging of new tweets with existing clusters.Sensitivity to noise (which is a known problem for document-pivot methods [21]) and fragmentation of clusters are drawbacks of this approach. Manual selection of trusted information providers and periodic defragmentation runs are needed to mitigate such effects.

The task of First Story Detection (FSD) discussed by Petrovic et al. [22] is closely related to document-pivot TDT. The goal is to detect the first document discussing a topic in a large corpus. A new story is created by a document having low similarity with all previously detected clusters. For fast retrieval of nearest neighbors for the incoming document locality sensitive hashing is used; however, such a solution is problematic when the nearest neighbors are not very similar to the query document.

B. Feature-pivot methods

Feature-pivot methods are closely related to topic models in natural language processing, namely statistical models to extract sets of terms that are representative of the topics occurring in a corpus of documents. Most state-of-the-art static topic models are based on Latent Dirichlet allocation (LDA) [23]. Even though LDA extensions for dynamic data have been proposed [24], alternative approaches trying to capture topics through the detection of keyword burstiness have been studied [25], mainly in the context of news media mining. The idea behind those methods is that breaking news, unlike other discussion topics, happen to reach a fast peak of attention from social media users as soon as they are publicly announced [26], [27]. Accordingly, the common framework that underlies most approaches in this category first identifies

bursty terms and then clusters them together to produce topic definitions.

Even before the diffusion of social media services, detection of bursty events had been studied in generic document sets. The method presented by Fung et al. [21], for instance, detects bursty terms by looking at where the frequency of the term in a given time window is positioned in the overall distribution of the number of documents containing that term. Once the bursty terms are found, they are clustered using a probabilistic model of cooccurrence. The need for a global topic term distribution restricts this approach to a batch mode of computation. Similar pipelines were tested for topic detection in social media (e.g., Twitter), but with additional emphasis on the enrichment of the obtained topics with non-bursty but relevant terms, URLs and locations [28].

Graph-based approaches detect keyword clusters based on their pairwise similarities. The algorithm by Sayyadi et al. [29] builds a term cooccurrence graph, whose nodes are clustered using a community detection algorithm based on betweenness centrality. Additionally, the topic description is enriched with the documents that are most relevant to the identified terms. Graphs of short phrases, rather than of single terms, connected by edges representing lexical inclusion or similarity have also been used [30]. Graph-based approaches have also been used in the context of collaborative tagging systems with the goal of discovering groups of tags pertaining to topics of social interest [31].

Alternative approaches based on signal processing have also been explored. Weng et al. [32] compute df-idf (a variant of tf-idf) for each term in each considered time slot, and apply wavelet analysis on consecutive blocks. The difference between the normalised entropy of consecutive blocks is used to construct the final signal. Bursty, relevant terms are extracted by computing the autocorrelation of the signal and heuristically determining a threshold to detect bursty terms. Also in this case, a graph between selected terms is built based on their cross-correlation and it is then clustered to obtain event definitions. The Discrete Fourier Transform is used by He et al. [33]: the signal for each term is classified according to its power and periodicity. Depending on the identified class, the distribution of appearance of a term in time is modeled using one or more Gaussians, and the KL-divergence between the distributions is then used to determine clusters.

When the additional information of the social network of the document producers is available, more sophisticated approaches are possible. In the method by Cataldi et al. [34] a PageRank-like measure is used to identify important users on the Twitter social network. Such centrality score is combined with a measure of term frequency to obtain a "nutrition" measure for each keyword. The trend of nutrition in time identifies bursty keywords. Clustering on a correlation graph of bursty keywords delineates the boundary of topics.

III. Topic detection from social media

Next, we define all the components of the topic detection pipeline. First (Section III-A), we present the problem statement and define some basic terminology. Then (Section III-B), we describe the data preprocessing and in the following sections we present six methods that take in as input the preprocessed data and output the detected topics.

A. Real-world events sensing: problem definition

We address the task of detecting topics in (near) real-time from social media streams. To keep our approach general, we consider that the stream is made of (usually short) pieces of text generated by social media users (*posts*, *messages*, or *tweets* in the specific case of Twitter). Posts are formed by a sequence of *words*, *terms* or *keywords* (we use the terms interchangeably in the following), and each one is marked with the *timestamp* of creation.

We address a user-centered scenario in which the user starts up the detection system by providing a set of *seed terms* that are used as initial filter to narrow down the analysis only to the posts containing at least one of the seed terms. Additionally, we assume that the *time frame* of interest (can be indefinitely long) and a desired *update rate* are provided (e.g., detect new trending topics every 10 minutes). The expected output of the algorithm is a *topic*, defined as a list of keywords, delivered at the end of each *time slot* determined by the update rate.

This setup fits well many real-world scenarios in which an expert of some domain has to monitor specific topics or events being discussed in social media [3], [35]. For instance, this is the case for computational journalism in which the media inquirer is supposed to have enough knowledge of the domain of interest to provide initial keywords to perform an initial filtering of the data stream. Even if it requires an initial human input, this framework still remains generic and suitable to any type of topic or event.

B. Data preprocessing

The content of user generated messages could be unpredictably noisy. To reduce the amount of noise before the proper topic detection is executed, the raw data extracted through the seed terms filter is subjected to three preprocessing steps.

- *Tokenization*. In a raw post, terms can be combined with any sort of punctuation and hyphenation and can contain abbreviations, typos, or conventional word variations. We use the Twokenizer tool [18] to extract bags of cleaner terms from the original messages by removing stopwords and punctuation, compressing redundant character repetitions, and removing mentions, i.e., IDs or names of other users included in the text for messaging purposes.
- *Stemming*. In information retrieval, stemming is the process of reducing inflected words to their root (or stem), so that related words map to the same stem. This process naturally reduces the number of words associated to each document, thus simplifying the feature space. In our experiments we use an implementation of the Porter stemming algorithm [36].
- *Aggregation*. Topic detection methods based on word or n-grams cooccurrences, or any other type of statistical inference, suffer in the absence of long documents. This is the case of social media, where user-generated content is typically in the form of short posts. In information

retrieval it is common practice to partially address this problem by concatenating different messages together to produce *super-documents* of larger size. We build super-documents based on two strategies. The first involves *temporal* aggregation that glues together N messages contiguous in time. The second involves *similarity-based* aggregation that attaches to a message all the near-duplicate messages posted in the same time slot, identified through an efficient document clustering method [22], which is also also used by one of the examined topic detection algorithms (see Section III-D).

Determining the effect of such preprocessing algorithms on the quality of the final topic is difficult to predict, and not much investigation about it has been done so far. For instance, the aggregation of posts in super-documents could on the one hand help to improve the word cooccurrence statistic but on the other hand introduces the risk of putting together terms related to different topics. In Section IV-C we will report the impact of the preprocessing on the results.

C. Latent Dirichlet Allocation

Topic extraction in textual corpora can be addressed through probabilistic topic models. In general, a topic model is a Bayesian model that associates with each document a probability distribution over topics, which are in turn distributions over words. Latent Dirichlet Allocation (LDA) [23] is the best known and most widely used topic model; we therefore use it as a baseline to compare our methods against. According to LDA, every document is considered as a bag of terms, which are the only observed variables in the model. The topic distribution per document and the term distribution per topic are instead hidden and have to be estimated through Bayesian inference. We use the Collapsed Variational Bayesian inference algorithm [37], an LDA variant that is computationally efficient, more accurate than standard variational Bayesian inference for LDA, and has parallel implementations already available in Apache Mahout[1]. LDA requires the expected number of topics k as a input and in our evaluation we explore the quality of the topic for different values of k (see Section IV-C). The estimation of the optimal k, although possible through the use of non-parametric methods [38], falls beyond the scope of this work.

D. Document-pivot topic detection (Doc-p)

The second method that we examine is an instance of a classical Topic Detection and Tracking method that uses a document-pivot approach. The flavour of the method is based on the work by Petrovic et al. [22], which uses locality sensitive hashing (LSH) in order to rapidly retrieve the nearest neighbour of a document and accelerate the clustering procedure. The principle behind this method is the same used for the near-duplicate detection in the similarity-based aggregation step of the preprocessing phase. It works as follows:

- Perform online clustering of posts: Compute the cosine similarity of the $tf\text{-}idf$ [39] representation of an incoming post to all other posts processed so far. If the similarity to the best matching post is above some threshold θ_{tf-idf}, assign the item to the same cluster as its best match; otherwise create a new cluster with the new post as its only item. The best matching tweet is efficiently retrieved by locality sensitive hashing.
- Filter out clusters with item count smaller than θ_n.
- For each cluster c, compute a score as follows:

$$score_c = \sum_{i=1}^{|Docs_c|} \sum_{j=1}^{|words_i|} \exp(-p(w_{ij}))$$

  where w_{ij} is the j^{th} term appearing in the i^{th} document of the cluster. The probability of appearance of a single term $p(w_{ij})$ is estimated from a reference dataset that has been collected from Twitter (see also Section III-E). Thus, less frequent terms contribute more to the score of the cluster.
- Clusters are sorted according to their score and the top clusters are returned.

The merit of using LSH is that it can rapidly provide the nearest neighbours with respect to cosine similarity in a large collection of documents. An alternative would be to use inverted indices on the terms that appear in the tweets and then compute the cosine similarity between the incoming document and the set of documents that have a significant term overlap with it; however, the use of LSH is much more efficient as it can provide the nearest neighbours with respect to cosine similarity directly.

In practice, for short posts such as tweets, we found that the similarity of two items is usually either close to zero or close to one (from around 0.8 to 1.0). This observation makes setting θ_{tf-idf} relatively easy: we set it to 0.5. Due to this phenomenon, items grouped together by this procedure are usually, but not always, near-duplicates (e.g., re-tweets). Therefore, it is clear that topics produced by this method will be fragmented, i.e. the same topic may be represented by different sets of near duplicate tweets. To begin dealing with this issue, we examine the use of different types of aggregation as described in Section III-B.

E. Graph-based feature-pivot topic detection (GFeat-p)

The next method is a first of a series of feature-pivot methods. Its unique feature is that for the feature clustering step it uses the Structural Clustering Algorithm for Networks (SCAN) [40]. A property of SCAN is that apart from detecting communities of nodes, it provides a list of hubs, each of which may be connected to a set of communities. In a feature-pivot approach for topic detection, the nodes of the graph would correspond to terms and the communities would correspond to topics. The detected hubs would then ideally be considered as terms that are related to more than one topic, something that would not be possible to achieve with a common partitional clustering algorithm and would effectively provide an explicit link between topics.

We select the terms to be clustered, out of the set of terms present in the corpus, using the approach in [18]. It uses an

[1] http://mahout.apache.org/

independent reference corpus consisting of randomly collected tweets. For each of the terms in the reference corpus, the likelihood of appearance $p(w|corpus)$ is estimated as follows:

$$p(w|corpus) = \frac{N_w + \delta}{(\sum_u^n N_u) + \delta n} \quad (1)$$

where N_w is the number of appearances of term w in the corpus, n is the number of term types appearing in the corpus and δ is a small constant (typically set to 0.5) that is included to regularize the probability estimate (i.e. to ensure that a new term that does not appear in the corpus is not assigned a probability of 0). To determine the most important terms in the new corpus, we compute the ratio of the likelihoods of appearance in the two corpora for each term. That is, we compute:

$$\frac{p(w|corpus_{new})}{p(w|corpus_{ref})} \quad (2)$$

The terms with the highest ratio will be the ones with significantly higher than usual frequency of appearance and it is expected that they are related to the most actively discussed topics in the corpus. Stop words, although already removed during preprocessing in our experiments, would typically have a ratio around 1. Once the high-ranking terms are selected, a term graph is constructed and the SCAN graph-based clustering algorithm is applied to extract groups of terms, each of which is considered to be a distinct topic. More specifically, the algorithm steps are the following:

- *Selection*: The top K terms are selected using the ratio of likelihoods measure and a node for each of them is created in the graph G.
- *Linking*: The nodes of G are connected using a term linking strategy. First, a similarity measure for pairs of terms is selected and then all pairwise similarities are computed. Various options for the similarity measure are explored: the number of documents in which the terms cooccur, the number of cooccurrences divided by the larger or smaller document frequency of the two terms, and Jaccard similarity. Moreover, either a kNN approach (linking each term with its k nearest neighbours) or an ϵ-based approach (link all pairs of nodes that have similarity higher that ϵ) can be used.
- *Clustering*: The SCAN algorithm is applied to the graph; a topic is generated for each of the detected communities.
- *Cluster enrichment*: The connectivity of each of the hubs detected by SCAN to each of the communities is checked and if it exceeds some threshold, the hub is linked to the community. A hub may be linked to more than one topic.

Clearly, the term linking step is crucial for the success of the method. Unfortunately, there is no straightforward method for determining the best similarity measure or node linking strategy to be used. Additionally, it can be expected that the graph construction parameters will need to vary for datasets with different topic granularities and levels of inter-topic connectivity. For this work, the parameters of graph construction were selected using the ground truth for a single independent timeslot. It should also be noted here that different parameters were required for the three different datasets (see Section IV-B), due to the fact that timeslots of different length were used for the three datasets and therefore there were large differences in the topic granularities and the inter-topic connectivity.

F. Frequent pattern mining (FPM)

A problem with feature-pivot methods like the one described in the previous section is that in order to group together a set of terms they only take into account their pairwise similarities which are based on some function of the number of cooccurrences between the pair of terms. In the case that there are closely interconnected topics that share a relatively large number of terms, this procedure is most likely to produce generic or merged topics. An option to deal with this issue is to take into account the simultaneous cooccurrence between more than two terms. This motivation leads naturally to consider the use of frequent itemset mining, a well-defined technique in transaction mining for topic detection to determine which *items* are likely to cooccur in a set of *transactions* [41].

In a social media context, an item is any term w mentioned in a post (excluding stop words, punctuation tokens, etc.). The transaction is the post, and the transaction set are all posts that occur in a time slot T_j. The number of times that any given set of terms occurs in the time slot is defined as its *support*, and any itemset that meets a minimum support is called a *pattern*. The initial challenge is to apply highly-scalable Frequent Pattern (FP) detection to each time slot in a large stream of posts and then rank the FPs in order to find the most relevant keyword sets for each time slot. These keyword sets may be considered as the topics that best illustrate the underlying social interactions. Below, we describe these two processing steps, FP detection and ranking.

1) FP detection: The FP-Growth algorithm is often used as a comparative baseline for frequent itemset mining, due to its good performance [42]. Our implementation uses a distributed version of the algorithm called Parallel FP-Growth [43] that is optimized for use on a Hadoop cluster. FP detection requires three rounds of Map-Reduce processing:

- Keyword list: For each time slot, the initial step of the FP-Growth algorithm is to create a list of keywords sorted by frequency. A minimum support is used to reduce the number of keywords being investigated.
- Parallel construction of an FP-tree data structure: For each time slot, an FP-Tree sorts the patterns according to their cooccurences and their support.
- Frequent pattern extraction: For each time slot, the parallel FP-tree structures are aggregated and analyzed to produce association rules on the transaction set in the form: $\{w_1, w_2\} \rightarrow P_i = \{w_3, w_4, ...\}$ with $support(P_i)$.

2) FP ranking: Once a set of frequent patterns has been extracted from the dataset, they are ranked and the top N results are returned as candidate topics. The challenge is to rank patterns such that keywords in the candidate topics are sufficiently related and with enough diversity to cover the different underlying subjects of conversation in the social interactions. A common way to rank patterns is to simply use the *support* of a given pattern; the more often a set

of keywords cooccurs, the more likely we can consider it relevant as a topic. Another measure of pattern relevance is the *lift*. It is defined as the ratio between the itemset support versus the expected frequency if the individual items were distributed independently. A higher lift for a pattern means that the keywords are more likely to be found together. The lift is appropriate to evaluate association rules, where one set of items imply the presence of another set.

Ranking by frequency favours short patterns, since a subset of any longer pattern is guaranteed to have the same or higher support. Ranking by lift favours longer patterns with cooccurrences of otherwise rare keywords. Another simple ranking mechanism to promote pattern length is to rank a pattern, then assign a *pattern length boost* weight for every additional token. Likewise, a *minimum pattern length* can be enforced by pruning smaller patterns. It is also interesting to note that removing punctuation tokens and stop words (the obvious non-keywords), if had not been carried out during pre-processing, may also be performed by ranking patterns with non-keywords by zero after the Parallel FP-Growth analysis. The results are equivalent; early pruning is the obvious choice for better performance in a running system, but late pruning permits more flexibility for investigating ranking functions, such as manually adding different "don't care" keywords that do not contribute to the underlying topics. This is particularly relevant when analyzing datasets obtained by monitoring specific seed keywords, which would otherwise overwhelm the detected frequent patterns.

It is important to note that pruning or penalizing a longer pattern (due to late stop word removal or specific keyword weighting) permits the subsets of that pattern to remain viable candidate topics. A subset of any pattern will always have equal or greater support, and there are always more subsets than there are larger patterns. However, when a longer pattern has subsets with exactly the same support (i.e. the subset of keywords only cooccur within the larger pattern), we can safely prune those subsets. Without additional information, we cannot tell which of the keywords in the larger pattern to discard in order to produce a better candidate topic. For instance, in the case of Twitter posts, the presence of long patterns is often due to retweeting popular status updates.

G. Soft frequent pattern mining (SFPM)

In Section III-F a frequent pattern mining approach for topic detection was developed. It provided an elegant solution to the problem of feature-pivot methods that take into account only pairwise cooccurrences between terms in the case of corpuses with densely interconnected topics. It can be said that it lies on the other end of the spectrum of methods that rely on the number of cooccurrences between terms: whereas the approach in Section III-E examined only pairwise cooccurrences, frequent pattern mining examines cooccurrences between any number of terms, typically larger than two. A question that naturally arises is if it is possible to formulate a method that lies between these two extremes. Such a method would examine cooccurrence patterns between sets of terms with cardinality larger that two, like frequent pattern mining does, but it would be less strict by not requiring that *all* terms in these sets cooccur frequently. Instead, in order to ensure topic cohesiveness, it would require that large subsets of the terms grouped together, but not necessarily all, cooccur frequently, resulting in a "soft" version of frequent pattern mining. In the following, we propose a method for achieving this.

The proposed approach works by maintaining a set of terms S, on which new terms are added in a greedy manner, according to how often they cooccur with the terms in S. In order to quantify the cooccurrence match between a set S and a candidate term t, we maintain a vector D_S for S and a vector D_t for the term t, both with length n, where n is the number of documents in the collection. The i^{th} element of D_S denotes how many of the terms in S cooccur in the i^{th} document, whereas the i^{th} element of D_t is a binary indicator that represents if the term t occurs in the i^{th} document or not. Thus, the vector D_t for a term t that frequently cooccurs with the terms in set S, will have a high cosine similarity to the corresponding vector D_S. Please note that some of the elements of D_S may have the value $|S|$, meaning that all items in S cooccur in the corresponding documents, whereas other may have a smaller value indicating that only a subset of the terms in S cooccur in the corresponding documents. For a term that is examined for expansion of S, it is clear that there will be some contribution to the similarity score also from the documents in which not all terms cooccur, albeit somewhat smaller compared to that documents in which all terms cooccur. This way we achieve the "soft" matching between a term that is considered for expansion and a set S. Finding the best matching term can be done either using exhaustive search or some approximate nearest neighbour scheme such as LSH.

As mentioned, we utilize a greedy approach that expands the set S with the best matching term, thus we need a criterion for terminating the expansion process. The termination criterion clearly has to deal with the cohesiveness of the generated topics, meaning that if not properly set, the resulting topics may either end up being too generic (with too few keywords) or really being a mixture of topics (with too many keywords related to possibly irrelevant topics). To deal with this we use the cosine similarity between S and the next best matching term. If the similarity is above some threshold, we add the term, otherwise the expansion process stops. This threshold is the only parameter of the proposed algorithm and is set to be a function of the cardinality of S. In particular we use a sigmoid function of the form:

$$\theta(S) = 1 - \frac{1}{1 + \exp((|S| - b)/c)} \tag{3}$$

The parameters b and c can be used to control the size of the term clusters and how soft the cooccurrence constaints will be. Practically, we set the values of b and c so that the addition of terms when the cardinality of S is small is easier (the threshold is low), but addition of terms when the cardinality is larger is harder. A low threshold for the small values of $|S|$ is required so that it is possible for terms that are associated to different topics and therefore occur in more documents rather than to ones corresponding to the non-zero elements of D_S to join

Algorithm 1 "Soft" frequent pattern mining for topic detection

```
T: The set of candidate terms
Topics = ∅
for each term t in T do
    S = t;
    D_S = D_t;
    ContinueExpanding = true;
    repeat
        t̂ = GetBestMatchingTerm(D_S, S, T);
        sim = CosineSimilarity(D_S, D_t̂);
        if sim > θ(S) then
            S = S ∪ t̂;
            D_S = D_S + D_t̂;
            for each D_S^i < |S|/2 set D_S^i = 0
        else
            ContinueExpanding = false;
        end if
    until ContinueExpanding
    Topics = Topics ∪ S
end for
Remove duplicates from Topics
```

the set S. The high threshold for the larger values of $|S|$ is required so that S does not grow without limit. Since we require a set of topics, rather than a single topic, the greedy search procedure is applied as many times as the number of considered terms, each time initializing S with a candidate term. This will produce as many topics as the set of terms considered, many of which will be duplicates, thus we post-process the results to remove these duplicates. To limit the search procedure in reasonable limits we select the top n terms with the highest likelihood-ratio (Eq. 2).

In early experiments with the described algorithm it was found that, after some time, especially if some very frequently occurring term has been added to the set, the vector D_S may include too many non-zero entries filled with small values. This may have the effect that a term may be deemed relevant to S because it cooccurs frequently only with a very small number of terms in the set rather than with most of them. In order to deal with this issue, after each expansion step, we reset to zero any entries of D_S that have a value smaller than $|S|/2$. In summary, the algorithm is presented in Algorithm 1.

Moreover, a nice feature of the approach is that the most relevant documents for a topic can be directly read from its vector D_S: these will be the ones with the highest document counts.

H. BNgram

Both the frequent itemset mining and soft frequent itemset mining approaches attempted to take into account the simultaneous cooccurences between more than two terms. However, it is also possible to achieve a similar result in a simpler way: use n-grams instead of unigrams. This naturally groups together terms that cooccur and it may be considered to offer a first level of term grouping. Using n-grams makes particularly sense for Twitter, since a large number of the status updates in Twitter are just copies or retweets of previous messages, so important n-grams will tend to become frequent.

Additionally, we introduce a new feature selection method. We take into account the changing frequency of terms over time as a useful source of information to detect emerging topics. The main goal of this approach is to find emerging topics in post streams by comparing the term frequencies from the current time slot with those of preceding time slots. We propose the $df\text{-}idf_t$ metric which introduces time to the classic $tf\text{-}idf$ score. We use historical data to penalize those topics which began in the past and are still popular in the present, and which therefore do not define new topics.

This approach indexes all keywords from the posts of the collection. The keyword indices, implemented using Lucene[2], are organized into different time slots. In addition to single keywords, the index also considers bigrams and trigrams. Once the index is created, the $df\text{-}idf_t$ score is computed for each n-gram of the current time slot i based on its document frequency for this time slot and penalized by the logarithm of the average of its document frequencies in the previous t time slots (see Equation 4).

$$df\text{–}idf_t = \frac{df_i + 1}{\log\left(\frac{\sum_{j=i}^{t} df_{i-j}}{t} + 1\right) + 1}. \tag{4}$$

In addition, a *boost* factor is considered to raise the importance of proper nouns (persons, locations and organizations, in our case) using a standard named entity recognizer [44], as they are essential keywords in most discussed stories. The use of this factor is similar to [17], where the authors highlight the importance of such words for grouping results. The selected values for this factor are based on the best values from the experiments of the previous work, being *boost*=1.5 in case the n-gram contains a named entity and *boost*=1 otherwise.

As a result of this process, a ranking of n-grams is created sorted by their $df\text{-}idf_t$ scores. A single n-gram is often not very informative, but a group of them often offers interesting details of a story. Therefore, we use a clustering algorithm to group the most representative n-grams into clusters, each representing a single topic. The clustering is based on distances between n-grams or clusters of n-grams. From the set of distances, those not exceeding a distance threshold are assumed to represent the same topic.

We define the similarity between two n-grams as the fraction of posts that contain both of them. We initially assign every n-gram to its own singleton cluster, then follow a standard "group average" hierarchical clustering algorithm [45] to iteratively find and merge the closest pair of clusters. When an n-gram cluster is joined to another, the similarities of the new cluster to the other clusters are computed as the average of the similarities of the combined clusters. The clustering is repeated until the similarity between the nearest un-merged clusters falls below a fixed threshold θ, producing the final set of topic clusters for the corresponding time slot.

In our experiments, we use a similarity threshold of $\theta = 0.5$ which means that two n-grams must appear in more than 50% of the same tweets in order to belong to the same topic. This assumption is stronger in our case because we are only considering the posts for a specific time-slot, so it is more likely that the n-gram clusters whose similarities are higher than the threshold represent the same topic. Preliminary experiments suggest that the value of θ is not critical.

[2]http://lucene.apache.org/core/

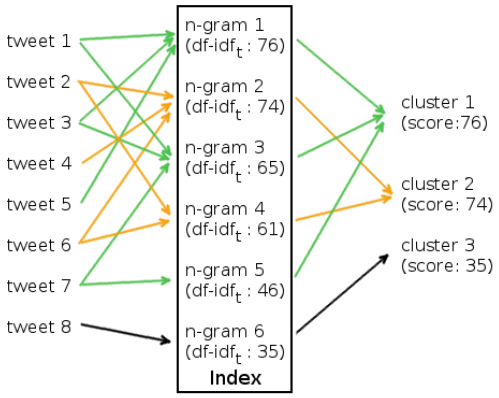


Fig. 1. Index organization where each n-gram keeps the references to tweets where it is contained. Every cluster is composed of different n-grams and its score is computed as the maximum df-idf_t value of them.

Finally, the clusters are ranked according to the highest df-idf_t score of the n-grams contained by the cluster as shown in Fig. 1. This ranking criterion is based on the assumption that each cluster score should be associated with the score of the most representative n-gram in the cluster, as the cluster is mainly composed of posts containing it.

Initial experiments (not described here) revealed that considered independently, the use of df-idf_t, n-grams and named entity boosting each improved the topic recall results, with the best results when all three are used.

The main contribution of this approach is the use of the temporal dimension of data to detect emerging stories. There are other similar approaches on term weighting considering the temporal dimension of data but most of them suffer from several shortcomings. For instance, Shamma et al. [25] present two methods of finding "peaky topics". They find the peak terms for a time-slot compared to the rest of the corpus, whereas we compare each slot to the immediately previous time slots. If some topic is discussed at several different times, their approach could miss this since the defining words would be highly frequent in the whole corpus. In addition, their approach only uses unigrams (i.e. single words) which often seem to be too limited to identify stories. Lastly, their use of the whole corpus favours batch-mode processing and is less suitable for real-time analysis.

IV. Experimental results

To test the validity of topic detection method proposed in Section III, we tested them on three Twitter datasets focused on three popular real-world events. We first present the datasets and describe the process of creating the ground truth. Then we present the performance of methods, comparing between different algorithm implementations.

A. Evaluation methodology

The evaluation framework consists of four steps:

Data collection. We extracted Twitter data from the public streaming API of Twitter[3]. Data collection was focused on three major events in 2012: the FA Cup final, the climax to the English domestic football season, the "Super Tuesday" (ST) primaries, part of the presidential nomination race of the US Republican Party, and the U.S. elections that took place in November 2012. Tweets related to those events where collected using a set of filter keywords and hashtags chosen by experts. We partitioned the datasets in timeslots, taking into account the average volume of tweets and the nature of the target events, specifically one hour for the ST, one minute for the FACup and ten minutes for the elections collection.

Extraction of ground truth. Clearly, there is a large number of topics hidden in the collections. However the sheer volume of the datasets implies that an overwhelming amount of effort would be required in order to manually extract the topics. Instead, we relied on mainstream media reports to identify significant topics and we focus on a subset of the actual set of topics. We reviewed the published media report accounts of the events and chose a set of stories that were significant, time-specific, and well-represented on news media in order to build a topic ground truth. For each topic, the ground truth consists of a set of *keywords* and a concise *headline* describing story. We assign a ground truth topic to one of the timeslots based on the time in which that topic emerged in mainstream media. Examples of some of the ground truth topics are shown in Table I.

Topic detection. We ran the topic detection algorithm on each timeslot for which at least one topic is contained in the ground truth. In total, we had 13 one-minute slots with at least one topic in FACup, eight one-hour slots for ST and twentysix ten-minute slots for the elections dataset. We used only the data from that slot as input to the methods. This choice is justified by the real-time scenario we are addressing: the topic detection system has to behave as a monitor by providing to the end-user the most recent trending topics.

Comparison of topic detection output with ground truth. The automatically detected topics (i.e., lists of keywords) were compared to the ground truth using three metrics:

- *Topic recall*: Percentage of ground truth topics that were successfully detected by a method. A topic was considered successfully detected in case the automatically produced set of keywords contained all mandatory keywords for it. To address the problem of spelling variations, we considered that a detected keyword matched a ground truth keyword if their Levenshtein similarity was above 0.8.
- *Keyword precision*: Percentage of correctly detected keywords (as described above) out of the total number of keywords for the topics that have been matched to some ground-truth topic in the time slot under consideration. The total precision of a method is computed by micro-averaging the individual precision scores over all time slots.
- *Keyword recall*: Percentage of correctly detected keywords over the total number of keywords of the ground truth topics that have been matched to some candidate topic in the time slot under consideration. The total recall is similarly computed by micro-averaging.

These scores were computed at the top n topics computed by

[3]https://dev.twitter.com/docs/streaming-apis

Event	Time range	Story	Keywords
FACup	5:26pm	Chelsea 1 - 0 Liverpool Ramires scores a goal from inside the box to the bottom left corner of the goal	ramires, goal, 1-0, chelsea, score, yes
	5:53pm	The referee shows Mikel a yellow card. Direct free kick taken by Daniel Agger	mikel, yellow, card, gerrard, foul, booking
	6:56pm	Liverpool nearly score Andy Carroll takes a shot. Petr Cech makes a fantastic save	andy,carroll, equalise, header, cech, line, over
Super Tuesday	7.00-7.30pm	Newt Gingrich: "Thank you Georgia! It is gratifying to win my home state so decisively to launch our March Momentum"	newt, gingrich, thank, georgia, march, momentum, gratifying
	8.00-8.20pm	NBC/CNN projects Mitt Romney will win the Massachusetts primary	nbc, cnn, project, mitt, romney, win, home, massachusetts, primary
	0.00-1.00am	Sarah Palin: "I voted for Newt Gingrich" on Fox	sarah, palin, voted, newt, gingrich, fox, cheerful
US Elections	9:00-9:10pm	Republican Party keeps control of the House of Representatives	GOP, republican, house, control
	10:30-10:40pm	Barack Obama wins Maine	Obama, wins, Maine
	1:40-1:50am	President Obama makes his victory speech	Obama, best, yet, come

TABLE I
EXAMPLES FROM THE TOPIC GROUND TRUTH. THE STORY AND TIME ARE TAKEN FROM OFFICIAL MEDIA SOURCES AND KEYWORDS ARE EXTRACTED FROM THE NEWS ARTICLES ACCORDINGLY.

the topic detection algorithms, for a range of values of n. They were automatically computed by an evaluation script, but to ensure the reliability of results we conducted several rounds of manual evaluation of results and confirmed their agreement with the automatically produced ones.

Note that we did not include topic precision as an evaluation measure. The reason is that to measure topic precision, we would need to compare the topics that our algorithms detect with the set of *every* newsworthy topic that took place at that particular time. A small missing cat and a national election may both be newsworthy in their own way, and people certainly send tweets about both, but there is no practical way to create a definitive list of all such events. Instead, we have only a subset of the topics that occurred in each timeslot, so we cannot be sure if the identified topics that have not been matched to the ground-truth topics are "genuine" topics or not. Thus, precision cannot be sensibly measured. One possibility would be a manual evaluation where the topics detected by each algorithm were subsequently labeled as actual or not actual topics by a human evaluator, who would need access to a complete archive of news events. Then it would be possible to compute topic precision. This would be extremely time-consuming, especially for studies such as this one which compares the efficiency of different algorithms in different types of datasets and therefore involves a very large number of runs.

B. Datasets

Next we describe the main features of the datasets used for the experiments. The three datasets and the ground truth we built for each of those are publicly available[4].

[4]http://www.socialsensor.eu/

1) FA Cup Final: The Football Association Challenge Cup, or FA Cup, is the main knock-out competition in English football and is the oldest association football competition in the world (being first held in 1871). In 2012, the two finalists were Chelsea and Liverpool. Chelsea won 2-1 with goals from Ramirez (11') and Drogba (52'). Carrol then scored for Liverpool (62'). The match lasted 90 minutes plus a 15 minute half-time break. It was the seventh time Chelsea won the FA Cup. Data was crawled using the official event hashtags, and the names of the teams and key players. The ground truth comprised 13 topics, including each of the three goals, some key bookings, and the start, middle and end of the match.

2) Super Tuesday primaries: In the US electoral system, the candidate for President for each political party is selected by a series of "primaries", which are elections held in individual states where members of the party vote for their choice of candidate. These primary elections take place from January to June in different states, and at the end of the process the candidate with most delegates elected by each state become the Presidential nominee. On some days, these primary elections take place in just one state, but on the first Tuesday in March, a large number of states hold their primary elections at the same time. Hence, Super Tuesday is usually the key moment when it is likely that the party nominee is selected.

Alaska, Georgia, Idaho, Massachusetts, North Dakota, Ohio, Oklahoma, Tennessee, Vermont and Virginia all voted on Super Tuesday 2012, Tuesday 6 March. In most states, voting took place from 7am to 7pm EST (12:00-2:00 GMT). The four Republican presidential candidates for 2012 were Mitt Romney, Ron Paul, Newt Gingrich, and Rick Santorum. Mitt Romney was considered the front runner, but Rick Santorum had been rising fast in the polls. Given the considerations above, the keyword list used for the data collection include the names and aliases of the four candidates, the ten states,

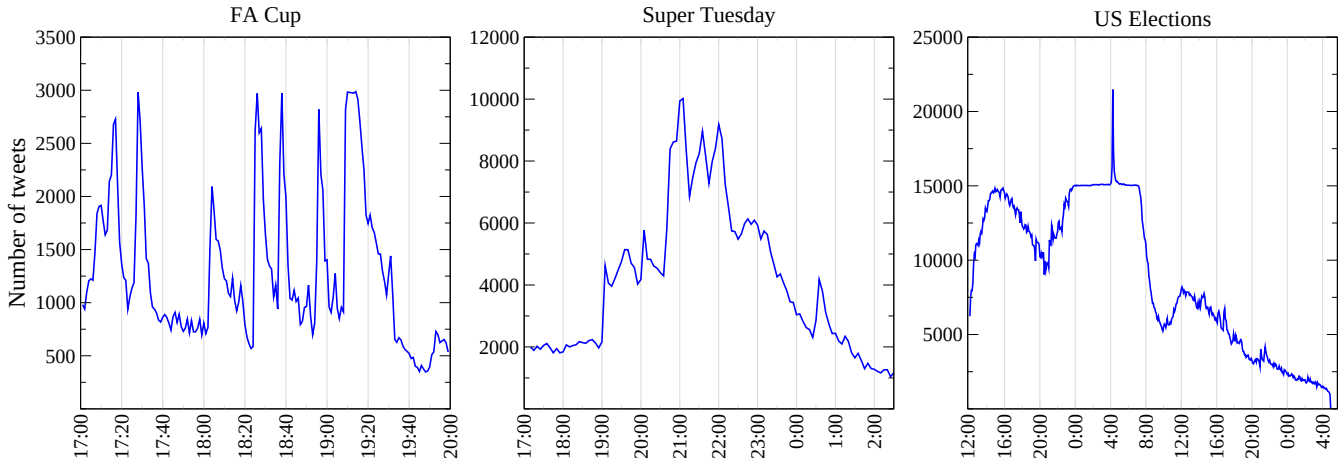


Fig. 2. Twitter activity during events. For the FA Cup, the peaks correspond to start and end of the match and the goals. For the two political collections, the peaks correspond to the main result announcements.

and the main news organizations reporting on the events (e.g., CNN and Fox).

The ground truth includes 22 topics covering stories such as the projection that a particular candidate would win a particular state and the televised speeches of several candidates. For evaluation purposes, we assigned each topic as belonging to a single hour. For example, if a real-world event happened at 22:45 ET, with corresponding Tweets occurring shortly afterwards, we assigned it to the 22:00-23:00 time slot.

3) US Elections: The United States presidential election of 2012 was held on Tuesday, November 6. President Barack Obama and his running mate, Vice President Joe Biden, were re-elected, defeating the Republican nominee, Mitt Romney, and his running mate, Paul Ryan. Each of the 50 US states returns a certain number of electoral college votes and each state declared their result independently over the course of the evening. There were also elections to both the US Senate and the House of Representatives and for several state governors. Some states also held referendums regarding issues such as same-sex marriage and the legalization of marijuana. The keyword list used for the crawl included the names of the candidates and various widely-used hashtags, such as #Election2012. The ground truth comprised 64 topics. The majority of these were the announcements by US television networks of the outcomes of the Presidential race in particular states, but also included referendum results, senate race results, and Obama's victory speech. For evaluation purposes, we assigned each topic to a 10-minute period.

The data collection process started several days before the beginning of all three events and ended some days after their completion. However, after examining the temporal pattern of the tweets, the datasets were trimmed to a narrower and more meaningful time interval. The activity profiles of the trimmed intervals are depicted in Figure 2. As the figures show, the topics considered in the case of ST are characterized by different durations, while the ones in FA Cup occur in very short intervals. The US Election attracted extremely high-levels of activity on Twitter. This lead to a saturation effect of the crawlers, and we collected very close to 3000 tweets per minute for large parts of the evening with an extra spike as the final outcome became clear. In total, we retrieved 474,109; 148,652; and 1,247,483 tweets respectively for ST, FA Cup and US election sets.

Collection	MSM tweets	Retweets of MSM
Super Tuesday	0.111%	5.96%
FA Cup	0.100%	1.44%
US Election	0.015%	1.21%

TABLE II
PROPORTION OF TWEETS SENT FROM MAINSTREAM MEDIA (MSM) ACCOUNTS AND RETWEETS OF MSM TWEETS

4) Preliminary datasets analysis: Here we examine some properties of the datasets, in order to assess their appropriateness for our topic detection task. First, we measure what proportion of the tweets collected were in fact relevant. To produce an estimate, we randomly sampled 200 tweets from each of the three sets and manually labelled each as relevant or not relevant. We found that 93.5% of the US Election tweets were relevant, 95% of the FA, and 89% of the ST set. While all these values are high enough to suggest we chose suitable keywords for our filters, it also suggests that the ST set may be less pure. By including the names of the participating states in our filter list, we inadvertently included tweets referring to sports events, holiday promotions etc. that happen to mention the state name.

A related question is to ask what proportion of the tweets was produced by mainstream media (MSM) outlets? We are using MSM descriptions of the events to define our ground truths, so if the majority of the tweets were themselves produced by MSM channels, then we may over-estimate the quality of our results. To investigate this, we identified the "official" Twitter accounts of the main news outlets (such as @CNN, @AP and @Reuters for the two political events, and @BBCSport and @ESPNTVUK for the football), while ignoring accounts of individual journalists, bloggers etc. We then counted a) the number of tweets sent directly from these accounts; and b) the number of MSM tweets retweeted by other accounts. Table II shows the average proportions for

each account. Clearly, very few of the tweets in our collections originated from these MSM accounts. It is worth noting that the US election featured an order of magnitude fewer MSM tweets than the other two collections; this is likely due to the MSM output being swamped by a huge number of tweets from many other sources during such a global and much-debated event. In all cases, it seems unlikely that the MSM tweets were dominant enough to have meant that our results were unduly biased towards the ground truth topics.

Additionally, we computed the entropy of the distribution of terms in each dataset. Intuitively, a higher entropy which directly indicates a more uncertain, wide distribution of terms in the corpus, implies a wider range of possible topics and therefore a more difficult topic detection task. The entropy of the distribution of terms for FA CUP was 10.73, the entropy for the Super Tuesday dataset was 12.11 and 11.76 for the U.S. elections dataset. This means that topic detection in the FA CUP dataset may be easier than in the other two datasets.

C. Results

First, we present the results for the methods with *no* preprocessing but the tokenization. Table III shows the precision and recall metrics in the case of a fixed number of topics N. Specifically, we selected $N = 2$ for the FACup and $N = 10$ for the political datasets to simulate a typical user-centered scenario where the user might want to receive a number of topics that is proportional with the breadth of the event. We observe that the BNgram method always achieves the best topic recall, always preserving a relatively good keyword precision and recall. The BNgram retrieves more topics than the other methods and the keywords appearing in such topics are pretty clean and well describe the topic.

The difference of performance with the other methods is slighter in the FACup case. In general, there is a noticeable difference of quality of detected topics between the three datasets. Recall of topics and keywords in FACup is almost always higher than the topic recall obtained in the political datasets, across all methods. Keyword precision is lower but comparable to the case of the political datasets. This is mainly due to the nature of the target event. Users commenting the match produce much more consistent content, since their attention is focused on a very narrow scope (the match itself) and for a limited time. Conversely, the stories about the primaries in US are plenty and interleaving and therefore more difficult to capture. This is supported empirically by the observation of higher entropy of terms used in the political discourse rather than in the tweets about the football match (as described in Section IV-B4). In particular, we notice that standard topic detection techniques such as LDA can perform reasonably well on very focused events while their performance can be dramatically low when considering more "noisy" events (no correct topics and keywords are detected in the Super Tuesday case).

To explore the variation of the performance when more topics are produced, we studied the performance metrics as the number N of top results considered varies. In particular, Figure 3 displays topic recall for the six different algorithms on the three datasets. Although BNgram clearly achieves the higher topic recall for smaller values of N for all datasets, its recall curve gets quickly flat because the number of topics it produces is always rather small. At higher values of N Doc-p and SFPM achieve better topic recall scores, especially in the elections dataset. Keyword precision and recall are very stable when N varies (not shown for brevity).

Table III, showing the results for the smallest values of N, indicate that for as strictly user-centered system, at which only the top few topics would be shown to the user, BNGram would be more useful than the other methods, as it achieves the highest topic recall score for all datasets. On the other hand, the most precise topic descriptions are achieved by FPM, for which K-Prec is usually the highest, whereas the most complete topic descriptions are achieved by either SFPM or LDA, for which K-Rec is highest in two and one datasets respectively.

The addition of preprocessing steps have also some impact on the retrieved topics. Surprisingly, we observe that the stemming step always deteriorates the results, lowering all the performance scores up to 21%. This is explained by the fact that stemming partially disrupts words association by merging too many words together. This effect seems to be very relevant for this task.

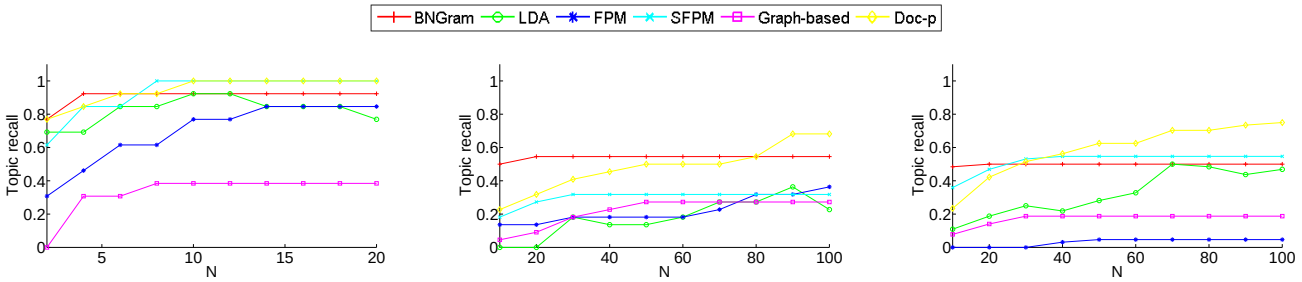
A different outcome is given by the aggregation step. As mentioned, aggregation may be a preprocessing option in cases of very short documents. Four different aggregation setups have been tested, in addition to the no-aggregation case. The first three involve time aggregation, where a number of subsequent tweets (10, 50 or 100) are merged in a single super-document. The fourth involves "topic aggregation", where tweets identified as near duplicated by a Doc-p procedure with a high threshold (0.95 in the experiments), are merged into a super-document. Topic recall for the six different algorithms for the different aggregation types is displayed in Figure 4.

The effect of the different types of aggregation depends on the targeted algorithm. In most cases the time aggregated datasets achieve lower topic recall scores than the non-aggregated and the topic-aggregated datasets. The larger the number of consecutive tweets that are aggregated, the lower the topic recall. On the other hand, topic aggregation seems to significantly improve topic recall for LDA and Doc-p, for which the topic recall scores are higher than those obtained by all other methods and any type of aggregation. Importantly, it is clearly observed that for Doc-p with topic aggregation (i.e. applying a two level document pivot method) assists in overcoming the problem of segmentation of topics produced by the plain Doc-p approach. Regarding the effect of different aggregation types on keyword precision and recall, it is observed that although keyword precision and recall do not change that much when the aggregation changes, keyword precision drops significantly for some of the algorithms.

The intuition behind the results is that aggregated tweets may represent a mixture of topics rather than a single topic, especially in the case of time aggregation, therefore they tend to introduce noisy associations of words. Therefore, it is likely that topics produced by a topic detection algorithm that does not explicitly consider a document as a mixture of topics, such

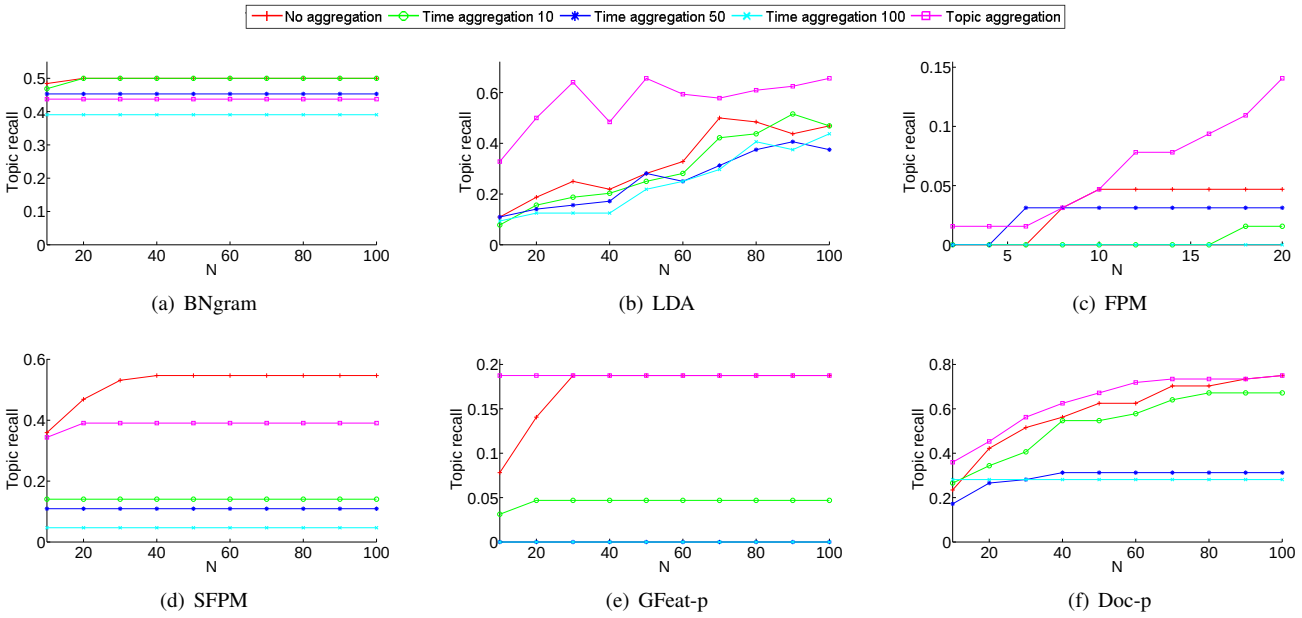
	FA Cup			Super Tuesday			US Elections		
Method	*T-REC@2*	*K-PREC@2*	*K-REC@2*	*T-REC@2*	*K-PREC@2*	*K-REC@2*	*T-REC@2*	*K-PREC@2*	*K-REC@2*
LDA	0.6923	0.1637	**0.6829**	0	0	0	0.1094	0.1654	0.6286
Doc-p	**0.7692**	0.3373	0.5833	0.2273	0.5116	0.6875	0.2344	0.4016	0.5862
GFeat-p	0	0	0	0.0455	0.3750	0.6000	0.0781	0.3750	0.4839
FPM	0.3077	**0.7500**	0.4286	0.1364	**1.0000**	0.4091	0	0	0
SFPM	0.6154	0.2336	0.6579	0.1818	0.4717	**0.8929**	0.3594	0.2412	**0.6953**
BNgram	**0.7692**	0.2989	0.5778	**0.5000**	0.6286	0.6471	**0.4844**	**0.4050**	0.5632

TABLE III
COMPARISON OF TOPIC DETECTION ALGORITHMS. T-REC, K-PREC, AND K-REC REFERS TO TOPIC-RECALL AND KEYWORD-PRECISION/RECALL RESPECTIVELY. BEST RESULTS ARE IN BOLD.



(a) FA Cup (b) Super Tuesday (c) US Elections

Fig. 3. Topic recall@N for the six different methods for the FACUP dataset (left), the Super Tuesday dataset (middle) and the U.S. elections dataset(right).



(a) BNgram (b) LDA (c) FPM

(d) SFPM (e) GFeat-p (f) Doc-p

Fig. 4. The effect of different pre-processing aggregation types on topic recall@N for the six different methods (using the U.S. elections dataset, similar results are observed for the other two datasets).

as LDA, to produce some topics which are in fact noisy. These topics will be represented by a larger number of keywords, therefore it is likely that keyword recall will be (at least) somewhat higher and keyword precision will be significantly lower as compared to then non aggregated case. On the other hand, the performance of LDA as indicated by all performance measures in our experiments, is not that affected much by time aggregation.

For illustrative purposes, in Table IV we present a set of randomly selected results produced by the BNgram method on both datasets. Each detected topic (set of keywords) is reported beside the corresponding story from the ground truth and a set of tweets that were retrieved by querying a full-text index of the collection with the topic keywords. In most cases, the detected topic is very well aligned with the textual description of the real-world story.

V. CONCLUSION

Topic detection from social media streams is a complex process that has to deal with all the interleaved dimensions that characterize the emergence of a story on a social network. The textual *content* of the user-generated posts, the distribution of

#	Detected topic	Corresponding story	Sample tweet
		FA Cup	
1	liverpool gets ambushed kalou defence box mazy run before @chelseafc great shoot #cfcwembley #facup	Salomon Kalou has an effort at goal from outside the area which goes wide right of the goal. Shot by Frank Lampard missed to the left of goal.	**@chelseafc:** Great mazy run by Kalou into the box but he gets ambushed by the Liverpool defence before he can shoot #CFCWembley #FACupFinal (SL)
2	mikel yellow card	Booking The referee shows Mikel a yellow card. Direct free kick taken by Daniel Agger..	**@chelseaindo:** 37mins Chelsea still lead 1-0. And yellow Card for Mikel
3	half time wembley chelsea lead liverpool final cup 1-0 through goal ramires early #cfc	Half time.	**@premierleague:** It's half-time in the FA Cup final at Wembley and Chelsea lead Liverpool 1-0 through an early goal from Ramires. #cfc #lfc #facupfinal
4	over line saved super cech claiming went @chelseafc carroll header liverpool #cfcwembley #facupfinal sl	Liverpool nearly score Andy Carroll takes a shot. Petr Cech makes a fantastic save.	**@chelseafc:** Carroll header is saved on the line by super Cech but Liverpool claiming it went over. #CFCWembley #FACupFinal (SL)
5	goal #cfcwembley #facupfinal sl @chelseafc chelsea	Didier Drogba scores.	**@chelseafc:** Chelsea goal #CFCWembley #FACupFinal (SL).
		Super Tuesday	
1	march momentum #marchmo #250gas launch win gratifying decisively @newtgingrich thank home state georgia	Newt Gingrich says Thank you Georgia! It is gratifying to win my home state so decisively to launch our March Momentum	**@newtgingrich:** Thank you Georgia! It is gratifying to win my home state so decisively to launch our March Momentum. #MarchMo #250gas #SuperTuesday
2	romney wins virginia republican presidential primary mitt @ap breaking	Fox/NBC is projecting Mitt Romney has won the Virginia primary.	**@ap:** BREAKING: Mitt Romney wins the Virginia Republican presidential primary. -RAS
3	romney wins idaho	NBC called Idaho (before all polls were closed.	**@nytimes:** NYT NEWS ALERT: Romney Wins Idaho Caucuses, A.P. Reports
4	romney mitt ohio primary #cnnelections won	AP has declared Ohio for Romney.	**@cnn:** CNN projects Mitt Romney has won the Ohio primary
5	kucinich concedes defeat news rep dennis marcy kaptur ohio democratic primary	US Rep. Dennis Kucinich concedes defeat to US Rep. Marcy Kaptur in Ohio Democratic primary.	**@ap:** BREAKING NEWS: US Rep. Dennis Kucinich concedes defeat to US Rep. Marcy Kaptur in Ohio Democratic primary.
		US Elections	
1	obama wins illinois projection	Obama projected to win Illinois by CNN	**@patrickdehahn:** CNN projects Obama wins in Delaware, Connecticut, DC, Illinois #election2012
2	democrat sherrod brown call wins senate seat ohio #election2012	Sherrod Brown is elected senator for Ohio according to AP	**@liberianjewels7:** @ap ap race call democrat sherrod brown wins senate seat ohio #election2012
3	wins arizona #election2012 romney	Mitt Romney won the state of Arizona according to several television networks	**@Violet_Oliver:** @ap ap race call romney wins arizona #election2012
4	legalized marijuana #election2012 washington @googlefacts	Washington state voted to end the prohibition of marijuana in Initiative 502	**@shayybabyxo:** @googlefacts washington legalized marijuana #election2012
5	@barackobama four more years	Several television networks report Obama has been re-elected; Obama tweeted "Four more years"	**@MessyNelle:** @barackobama four more years http://t.co/6ortbfqt

TABLE IV
EXAMPLE RESULTS AUTOMATICALLY DETECTED BY THE BNGRAM METHOD.

the messages in *time* and the nature of the *events* around which the crowd is commenting are the three most important aspects to consider. Given that no standard topic detection technique has been established yet, comparative analysis are needed to understand to what extent these dimensions determine the quality of the detected topics.

We compare six different topic detection algorithms –two baselines from the literature and four novel methods– by testing them on Twitter data streams about three real-world events and we match the automatically generated topics with a reliable ground truth from mainstream media, that allows us to get quantitative measures about the topic reliability. We produced and evaluated topics at different stages of the event in order to capture the evolving stories related to it. All the algorithms leverage the content dimension with different approaches, ranging from the analysis of cooccurrence of unigrams to n-cooccurrences of unigrams, up to cooccurrences of n-grams. The method based on n-grams outperforms the others, suggesting that more complex aggregation of keywords better capture the ground truth topic. Orthogonally, we explore the time dimension by proposing a time-dependent ranking (namely $df\text{-}idf_t$) to boost the importance of bursty events. We also show the impact that standard preprocessing steps such as stemming and aggregation of documents in super-documents can affect the topic detection outcome. Finally, we give insights about the role that the dataset type (i.e., the type of target event) can have on detected topics. We find that classic topic models such as LDA can well capture the stories happening during events with narrow topical scope, while for broader events, where many different stories can run in parallel at the same time, methods based on n-grams cooccurrence plus time-dependent boost are much more suitable.

Several further directions can be explored, including the impact on the detection output of other orthogonal dimensions such as the social network between the content generators. Furthermore, an extension of proposed methods could be able to detect most interesting topics occurring within the event, thus enabling to notify only the most relevant stories

happening. Lastly, it could be interesting to study the effects of using n-grams, df-idf_t and Named Entity boosting in isolation.

REFERENCES

[1] A. Panisson, "Visualization of Egyptian revolution on Twitter," http://slashdot.org/story/11/02/15/1544254, Feb 2011.

[2] M. Wall and S. E. L. Zahed, ""I'll Be Waiting for You Guys": A YouTube Call to Action in the Egyptian Revolution," *Journal of Communication*, vol. 5, pp. 1333–1343, 2011.

[3] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: real-time event detection by social sensors," in *WWW: 19th ACM International Conference on World Wide Web*. ACM, 2010, pp. 851–860.

[4] M. D. Conover, B. Gonçalves, J. Ratkiewicz, A. Flammini, and F. Menczer, "Predicting the Political Alignment of Twitter Users," in *SocialCom: 3rd IEEE International Conference on Social Computing*, Boston, Massachusetts, USA, Oct 2011.

[5] R. Kumar, J. Novak, and A. Tomkins, "Structure and Evolution of Online Social Networks," in *KDD: 12th ACM International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2006, pp. 611–617.

[6] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *KDD: 14th ACM International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2008, pp. 462–470.

[7] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the Flickr social network," in *WWW: 18th ACM International Conference on World Wide Web*. New York, NY, USA: ACM, 2009, pp. 721–730.

[8] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Trans. Web*, vol. 6, no. 2, pp. 9:1–9:33, Jun. 2012.

[9] S. Papadopoulos, A. Vakali, and I. Kompatsiaris, "The dynamics of content popularity in social media," *International Journal of Data Warehousing and Mining*, vol. 6, no. 1, pp. 20–37, 2010.

[10] S. Cohen, J. T. Hamilton, and F. Turner, "Computational journalism," *Comm. of the ACM*, vol. 54, pp. 66–71, Oct 2011.

[11] "New York Times Cascade Project - nytlabs.com/projects/cascade.html," 2011.

[12] D. Quercia, J. Ellis, L. Capra, and J. Crowcroft, "Tracking "Gross Community Happiness" from Tweets," in *CSCW: ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 2012, pp. 965–968.

[13] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, "Detecting and Tracking Political Abuse in Social Media," in *ICSWM: 5th International AAAI Conference on Weblogs and Social Media*, 2011.

[14] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The socialbot network: When bots socialize for fame and money," in *ACSAC: 27th Annual Computer Security Applications Conference*. New York, NY, USA: ACM, 2011, pp. 93–102.

[15] L. M. Aiello, M. Deplano, R. Schifanella, and G. Ruffo, "People are Strange when you're a Stranger: Impact and Influence of Bots on Social Networks," in *ICWSM: 6th AAAI International Conference on Weblogs and Social Media*. AAAI, 2012, pp. 10–17.

[16] J. Allan, Ed., *Topic detection and tracking: event-based information organization*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[17] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in Twitter," *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, vol. 3, pp. 120–123, 2010.

[18] B. O'Connor, M. Krieger, and D. Ahn, "TweetMotif: Exploratory Search and Topic Summarization for Twitter," in *ICWSM*, W. W. Cohen, S. Gosling, W. W. Cohen, and S. Gosling, Eds. The AAAI Press, 2010.

[19] H. Becker, M. Naaman, and L. Gravano, "Beyond Trending Topics: Real-World Event Identification on Twitter," in *ICWSM: 5th International AAAI Conference on Weblogs and Social Media*, 2011.

[20] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "TwitterStand: News in Tweets," in *GIS: 17th ACM International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM, 2009, pp. 42–51.

[21] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *VLDB: 31st International Conference on Very Large Data Bases*. VLDB Endowment, 2005, pp. 181–192.

[22] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming First Story Detection with Application to Twitter," in *HLT: Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 181–189.

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar 2003.

[24] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *ICML: 23rd International Conference on Machine Learning*. New York, NY, USA: ACM, 2006, pp. 113–120.

[25] D. A. Shamma, L. Kennedy, and E. F. Churchill, "Peaks and persistence: Modeling the Shape of Microblog Conversations," in *CSCW: ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 2011, pp. 355–358.

[26] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *WSDM: 4th ACM international conference on Web search and data mining*. New York, NY, USA: ACM, 2011, pp. 177–186.

[27] J. Lehmann, B. Gonçalves, J. J. Ramasco, and C. Cattuto, "Dynamical Classes of Collective Attention in Twitter," in *WWW: 21st ACM International Conference on World Wide Web*. New York, NY, USA: ACM, 2012, pp. 251–260.

[28] M. Mathioudakis and N. Koudas, "TwitterMonitor: Trend Detection over the Twitter Stream," in *SIGMOD: International Conference on Management of Data*. New York, NY, USA: ACM, 2010, pp. 1155–1158.

[29] H. Sayyadi, M. Hurst, and A. Maykov, "Event detection and tracking in social streams," in *ICWSM*, E. Adar, M. Hurst, T. Finin, N. S. Glance, N. Nicolov, and B. L. Tseng, Eds. The AAAI Press, 2009.

[30] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-Tracking and the Dynamics of the News Cycle," in *KDD: 15th ACM International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2009, pp. 497–506.

[31] S. Papadopoulos, Y. Kompatsiaris, and A. Vakali, "A graph-based clustering scheme for identifying related tags in folksonomies," in *DaWaK: 12th International Conference on Data Warehousing and Knowledge Discovery*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 65–76.

[32] J. Weng and B.-S. Lee, "Event Detection in Twitter," in *5th International Conference on Weblogs and Social Media*. The AAAI Press, 2011.

[33] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *SIGIR: 30th Annual International ACM Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2007, pp. 207–214.

[34] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation," in *MDMKDD: 10th International Workshop on Multimedia Data Mining*. New York, NY, USA: ACM, 2010, pp. 4:1–4:10.

[35] S. Diplaris, G. Petkos, S. Papadopoulos, Y. Kompatsiaris, N. Sarris, C. Martin, A. Goker, D. Corney, J. Geurts, Y. Liu, and J.-C. Point, "SocialSensor: Surfacing real-time trends and insights from multiple social networks," in *In Proceedings of the 2012 NEM Summit*, Oct 2012, pp. 47–52.

[36] M. F. Porter, "Readings in Information Retrieval," K. Sparck Jones and P. Willett, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, ch. An algorithm for suffix stripping, pp. 313–316.

[37] Y. W. Teh, D. Newman, and M. Welling, "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation," in *Advances in Neural Information Processing Systems*, vol. 19, 2007.

[38] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[39] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1986.

[40] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "SCAN: a Structural Clustering Algorithm for Networks," in *KDD: 13th ACM International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2007, pp. 824–833.

[41] B. Goethals, *Frequent Set Mining*, 2005, pp. 377–397.

[42] C. Győrödi and R. Győrödi, "A Comparative Study of Association Rules Mining Algorithms," 2004.

[43] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang, "PFP: Parallel FP-growth for Query Recommendation," in *RecSys: ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2008, pp. 107–114.

[44] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling," in *ACL: 43rd Annual Meeting on Association for Computational Linguis-*

tics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 363–370.

[45] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *The Computer Journal*, vol. 26, no. 4, pp. 354–359, 1983.