A soft frequent pattern mining approach for textual topic detection

Georgios Petkos
CERTH-ITI
Thessaloniki, Greece
gpetkos@iti.gr

Symeon Papadopoulos
CERTH-ITI
Thessaloniki, Greece
papadop@iti.gr

Luca Aiello
Yahoo!
Barcelona, Spain
alucca@yahoo-inc.com

Ryan Skraba
Google
Paris, France
ryan@skraba.com

Yiannis Kompatsiaris
CERTH-ITI
Thessaloniki, Greece
ikom@iti.gr

ABSTRACT

Textual topic detection methods that work by clustering terms according to their cooccurrence patterns are called feature-pivot methods. Typically, the similarity measure that is used for such clustering methods takes into account the cooccurrence patterns of only pairs of items. In this work, we argue that examining the *simultaneous cooccurrence patterns of a larger number of terms*, is a better option when the corpus contains a set of closely related fine-grained topics. To this end, we treat the topic detection problem as a Frequent Pattern Mining problem and propose a novel algorithm for "soft" Frequent Pattern Mining. We test the proposed approach using three annotated datasets collected from Twitter and compare it to a set of algorithms that includes a graph-based feature-pivot approach that takes into account only cooccurrence patterns, a standard Frequent Pattern Mining algorithm and Latent Dirichlet Allocation. The results indicate that SFPM is performing better than the other tested methods and show a clear improvement over the standard FPM approach.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous; I.5.4 [**Pattern Recognition**]: Application

General Terms

Algorithms, Theory, Experimentation

Keywords

Topic detection, Frequent pattern mining, Soft frequent pattern mining, Feature-pivot

WIMS '14, June 02 - 04 2014, Thessaloniki, Greece
Copyright 2014 ACM 978-1-4503-2538-7/14/06 ...$15.00.
http://dx.doi.org/10.1145/2611040.2611068

1. INTRODUCTION

A wealth of textual data is constantly being generated on the Web, in the form of news articles, blog posts, tweets, etc. This data is a very rich source of information that relates to real-world activity and thus may be of interest to diverse audiences. Consequently, the development of text-processing and topic detection techniques for the extraction of insights from large amounts of online text has attracted significant research effort.

Textual topic detection methods largely fall in three classes. The first, termed **document-pivot** methods, group together individual documents according to their similarity. The second class, termed **feature-pivot** methods, group together terms according to their cooccurrence patterns. The third class, **probabilistic topic models**, treats the problem of topic detection as a probabilistic inference problem. Naturally, methods belonging to each of these classes represent topics differently. A document-pivot method represents a topic with sets of relevant documents; a feature-pivot method represents topics with a set of terms and a probabilistic topic model represents a topic by a distribution of terms. *In this work, we focus on feature-pivot methods.* In particular, we examine the effect of the "degree" of examined cooccurrence patterns on the term clustering procedure. Previous approaches typically utilized a similarity metric between *pairs of terms*, i.e. the examined cooccurrence patterns has a degree of two. We argue that in the case that the corpus contains a set of related fine-grained topics, examining only pairwise cooccurrence patterns is likely to produce mixtures of topics instead of the individual fine-grained topics. In this work, in order to deal with this issue, we propose to examine simultaneous cooccurrence patterns between more than two terms, i.e. we look at cooccurrence patterns of degree larger than two. We build on the concept of Frequent Pattern Mining (FPM) and propose the Soft Frequent Pattern Mining (SFPM) algorithm. Whereas a pure FPM formulation for topic detection [10] would look for sets of items (of any size) that cooccur frequently in the same document, the SFPM approach is a less strict version of FPM, by not requiring that all terms in a set frequently cooccur together but large subsets of the set do.

SFPM is evaluated on three tasks. All tasks involve the detection of specific topics on a collection of tweets that are related to specific events, the USA Super Tuesday primaries on March 2012, the FA Cup final in May 2012 and the US elections in November 2012. SFPM is evaluated against a graph-based feature-pivot approach that examines the pairwise cooccurrence of terms, a standard FPM approach and a commonly used probabilistic topic detection algorithm, Latent Dirichlet Allocation (LDA). The results indicate that SFPM is performing better than competing methods and show a clear improvement over the standard FPM approach.

The rest of the paper is structured as follows. Section 2 presents a review of topic detection methods. Subsequently, Section 3 discusses the problem with feature-pivot methods that take into account only pairwise cooccurrence patterns and introduces the SFPM algorithm. Section 4 presents the experiments conducted in order to evaluate SFPM against competing methods. Finally, Section 5 concludes the paper and discusses some future work.

2. RELATED WORK

As previously discussed, there are three large classes of topic detection methods: document-pivot, feature-pivot, and probabilistic topic models. It is not easy to conclude on which class produces the best results. Nevertheless, it has been reported [8] that the similarity of pairs of documents may easily be dominated by noisy features and therefore items may be incorrectly clustered using document-pivot methods. Probabilistic approaches have been reported to produce good results, however, they are typically computationally expensive. In the following, we present a short review of topic detection approaches that belong to the three aforementioned classes.

2.1 Document-pivot approaches

Document-pivot approaches typically compute a measure of similarity between either a pair of documents or a document and a prototype cluster representation. In the first case, if the similarity between the incoming document and the best matching document that is already in the collection is above some threshold, then the incoming document is added to the same cluster as the best matching document. If not, a new cluster is generated. Similarly, in the latter case, if the similarity of the incoming document to the best matching cluster is above some threshold, the item is added to the cluster, otherwise a new cluster is created. The document-pivot approaches appearing in the literature generally apply a variation of one of these two approaches. Typically they differ in that they compute the similarity in different ways, they apply special techniques for finding the best matching item or cluster, or they use some post-processing step.

For instance, [19] presents a typical document-pivot method that has been applied on the problem of breaking news detection. In this approach, tweets are represented using the classical bag of words representation and a TF-IDF weighting scheme. The similarity measure used for clustering is a combination of document-to-document and document-to-cluster. In particular, an incoming tweet is clustered with other tweets by comparing its TF-IDF vector with the TF-IDF vector of the first item and the TF-IDF vector of the most common terms in each cluster. A similar approach is presented in [22]. It utilizes a variant of the incremental clustering approach, termed "leader-follower" clustering, which takes into account both the textual and the temporal proximity between an incoming tweet and each cluster. A feature of the approach that makes it appropriate for large datasets is that, not all existing clusters are examined, but only the ones which share at least one textual feature with the incoming tweet, using a simple indexing scheme. Moreover, clusters older than some limit are not examined.

Another incremental document-pivot approach that attempts to deal with scalability issues, albeit one that is based on document-to-document similarity, can be found in [18]. It utilizes a modification of Locality Sensitive Hashing (LSH) to retrieve fast the best match for each incoming document.

In general, such incremental clustering approaches require appropriate setting of a threshold parameter. If the threshold is set too low, then a single cluster will either represent too generic topics or mixtures of topics. If the threshold is set too high, then clusters will tend to be fragmented and a single topic will be represented by many clusters. The threshold is in most cases set empirically and also it clearly depends on the selected similarity measure. Moreover it should be noted that fragmentation is a more general issue for document-pivot approaches. That is, it is likely that due to the fact that a topic may be expressed in different ways, many clusters represent a single topic. However, since with time the representation of a topic becomes enriched, it is possible that a merging procedure can be applied. Various approaches attempt to deal with fragmentation. For instance, [22] periodically attempts to find fragmented clusters and labels the older as the slave cluster and the newer as the master cluster. If a new item is clustered with the slave cluster it is automatically assigned to the master cluster. In order to deal with fragmentation, a second pass of the incremental clustering procedure is applied in [2].

2.2 Feature-pivot methods

Feature-pivot methods attempt to cluster together terms according to their cooccurrence patterns. In general, a first step of such approaches involves the selection of the terms that will be clustered and for which the cooccurrence patterns will be computed. Different criteria may be used. For instance, the most common terms that are not stop words or the most bursty terms. In the second step, some form of inter-term similarity, typically between pairs of terms is computed and is used in conjunction with some clustering procedure. Approaches that have appeared in the literature present a large variety of ways to select the set of terms to be clustered, to compute the inter-term similarity and to perform clustering.

One of the most interesting feature-pivot approaches is presented in [6]. This approach is interesting in that it utilizes social features in order to select the terms to be clustered. It defines the "energy" of each term, a quantity that takes into account both the frequency of term occurrence but also the importance of the users that have posted documents including the term. The terms with the highest "energy" are clustered using a graph-based algorithm. That is, a graph where each node represents a term and each edge is weighted by some measure of inter-term correlation is formed. An adaptive edge-thinning algorithm is used in conjunction with a reachability algorithm to obtain a set of term clusters.

Typically though, the set of terms to be clustered is selected by taking into account only the frequency or the burstiness of the appearance of terms. For instance, in [8] terms are selected according to their burstiness. This is done by modelling the distribution of the number of documents that contain each term in a single window of time. For some window in time, a feature will be termed bursty if the number of documents that contain it is above the median of the distribution. Once the bursty features are found, the features are clustered using a probabilistic model which takes into account the cooccurence of features.

Of particular interest are approaches that are based on signal processing operations. One of them is "Event Detection with Clustering of Wavelet-based Signals" (EDCoW) [24]. This computes the Document Frequency - Inverse Document Frequency (DF-IDF, similar to TF-IDF) for each term and for each time slot. EDCoW forms a signal with all the DF-IDF values for each term and transforms it using a sequence of signal processing operations. The terms to be clustered are the ones with the highest autocorrelation value for the corresponding final signal. Eventually, the cross-correlation between all pairs of the selected features is computed and this is used to form a correlation matrix which is essentially a representation of a graph with nodes that correspond to the features and edge weights that correspond to the correlations. A simple heuristic rule that takes into account the distribution of the cross-correlation values is used to sparsify the matrix. Eventually, an eigenvector-based, spectral algorithm is used to cluster the graph's nodes.

Yet another approach that utilizes signal processing techniques can be found in [13]. This also constructs a signal for each term using DF-IDF values. However, it applies the Discrete Fourier Transform on it and determines the dominant period and power spectrum. These values are used to select the terms that will be clustered. Eventually, pairwise similarities between pairs of the selected terms that are based on the KL-divergence on the distributions of appearance of the terms are computed. A simple cost function that depends on the KL-divergence is defined and a greedy algorithm is used to group together the selected terms. One interesting aspect of this method is that it is able to characterize topics as periodic or aperiodic.

An interesting graph-based approach is presented in [23]. This organizes the selected terms (those with document frequency above some threshold) in the "KeyGraph". The "KeyGraph" has one node for each selected term and connects terms if they cooccur more than a prespecified number of times and if the conditional probability of the one appearing provided that the other has also appeared is above some other threshold. The clustering algorithm progressively removes the edges with the highest betweeness centrality and this results in parts of the graph to separate from each other. Additionally, the approach is able to handle terms that may be relevant to more than one topic by conditionally duplicating nodes during the clustering procedure.

To summarize, most of the feature-pivot methods in the literature, regardless of the employed term selection mechanism, examine the cooccurrence patterns (in various forms) between pairs of terms. In practice, depending on the clustering algorithm, this may lead to common terms being incorrectly grouped with some terms, simply due to the fact that they are quite common and they may be marginally linked to a large number of topics. In the next Section, we will examine, how taking into account cooccurrence patterns with degree larger than two can assist in topic detection.

2.3 Probabilistic topic models

The third class of topic detection models comprises probabilistic topic models. These approaches represent the joint distribution of topics and terms using a generative probabilistic model, which consists of a set of latent variables that represent topics, terms, hyperparameters, etc. Two very well known probabilistic topic models that have been extended in many ways are Probabilistic Latent Semantic Analysis (PLSA) [14] and Latent Dirichlet Allocation (LDA) [4]. LDA is probably the most popular probabilistic topic model; it uses hidden variables that represent the per-topic term distribution and the per-document topic distribution. Learning and inference in LDA is typically performed using Variational Bayes [7] but other approaches such as using Gibbs sampling have appeared [25]. Additionally, supervised versions of LDA have surfaced, such as Labeled-LDA [21] (with an application on Twitter appearing as TweetLDA in [20]). LDA will be used as a baseline for the approach proposed in this paper. A review of probabilistic topic models can be found in [3].

3. BEYOND PAIRWISE COOCCURRENCE ANALYSIS

Feature-pivot methods discover topics by putting together sets of keywords based on their cooccurrence patterns. Most feature-pivot methods in the literature examine the cooccurrence patterns between pairs of terms. In the case that there are many related topics in the corpus, it may not be possible to distinguish these topics by examining only pairwise cooccurrence patterns. For instance, in one of the datasets that we examine, we focus on a political event (Super Tuesday) and there are topics like "Mitt Romney wins Virginia", "Romney appears on TV and thanks Virginia", "Newt Gingrich gives a speech on healthcare" and "Newt Gingrich wins Vermont". In these topics there is a set of dominant terms ("wins", "Romney", "Virginia" and "Gingrich"), each of which cooccurs with more than one other term; e.g., "wins" cooccurs with "Romney", "Gingrich", "Virginia" and "Vermont" and at the same time these terms cooccur with each other in a number of documents. Thus, if we examine only patterns of pairwise cooccurrences it is likely that all such terms end-up being grouped together and the obtained topics are mixed topics that unclearly represent all these finer topics together. Thus, taking into account only pairwise cooccurrence patterns in cases that fine-grained topic detection is required may lead to low quality results. On the other hand, it is clear that if we instead examine the simultaneous cooccurrence patterns in the same document between a larger number of terms (e.g. "wins", "Romney" and "Virginia" at the same time) we may be able to identify such finer topics.

3.1 FPM for topic detection

We now proceed to examine approaches that move beyond pairwise cooccurrence analysis. A straightforward way to consider simultaneous cooccurrence patterns between more than two terms is to apply FPM techniques [9]. FPM involves a set of techniques that were developed to discover frequent patterns in a large database of transactions. In the context of feature-pivot methods, one would look for terms that frequently occur together. Considering that there is a huge number of possible sets of terms that may frequently occur together, appropriate algorithms such as Apriori [1], DIC [5], DHP [17] and FP-Growth [12] have been developed to efficiently discover such patterns. For a review of FPM

Algorithm 1 SFPM for topic detection

```
Input: $C$: A corpus of $n$ documents
  $K$: Number of top terms to be selected
  $b, c$ : The parameters of the sigmoid
Output: $Topics$: The set of resulting topics
  $T = SelectTopTerms(C, K)$
  for each term $t$ in $T$ do
    $D_t = ComputeOccurrenceVector(C, t)$
  end for
  $Topics = \emptyset$
  for each term $t$ in $T$ do
    $S = t$;
    $D_S = D_t$;
    $ContinueExpanding = true$;
    repeat
      $\hat{t} = GetBestMatchingTerm(D_S, S, T)$;
      $sim = CosineSimilarity(D_S, D_{\hat{t}})$;
      if $sim > \theta_{b,c}(S)$ then
        $S = S \cup \hat{t}$;
        $D_S = D_S + D_{\hat{t}}$;
        for i=1,...,n do
          if $D_S^i < |S|/2$ then
            $D_S^i = 0$
          end if
        end for
      else
        $ContinueExpanding = false$;
      end if
    until $ContinueExpanding$
    $Topics = Topics \cup S$
  end for
  Post-processing (duplicate removal)
```

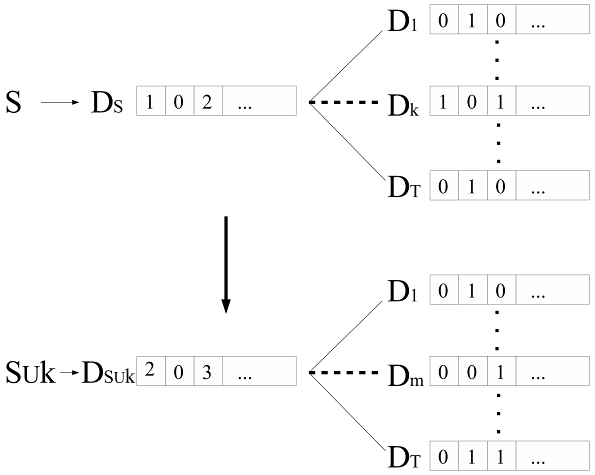

Figure 1: A single expansion of the set S. The best match D_k of the D_S vector is found, the set S is expanded with the element k and D_k is added to D_S.

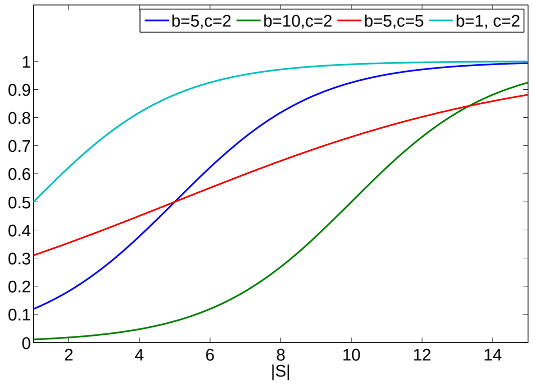

Figure 2: Different shapes of the sigmoid depending on the b and c parameters. The values $b = 5$ and $c = 2$ are used in this work.

methods please see [11].

An FPM method for topic detection that utilizes the FP-stream algorithm has appeared in [10]. Interestingly, FPM has also been used in conjunction with probabilistic topic models in [15], in order to enrich the representation of documents before they are processed by standard probabilistic topic models. That is, the representation of documents is enriched with the patterns that involve the terms in the document and that have a high support. The rationale is that such patterns convey semantic relationships between terms that may be missed if only the terms in the document are considered. FPM has also been used in a similar manner to improve the document retrieval in [27].

In this work we also experiment with a FPM approach. We utilize the FP-Growth algorithm, which is based on the use of the FP-Tree structure. Once the FP-Tree is constructed, all patterns with some minimum support are extracted from it (support is the number of transactions/documents that contain the pattern) and are ranked. The top patterns are returned from the algorithm. As mentioned later, we experiment with datasets from Twitter: we treat each tweet as a transaction, whose items are the terms appearing in the tweet (all terms and hashtags are treated in the same manner).

3.2 Proposed approach: SFPM

It can be said that FPM lies on one end of the spectrum of methods that examine cooccurrence patterns between terms: whereas a typical feature-pivot method examines only pairwise cooccurrences, FPM examines the simultaneous cooccurrence between any number of terms, typically larger than two. As argued before, we are interested in cooccurrence patterns of degree larger than two in order to capture closely related topics; however, in some cases it could be that due to the strict requirement of FPM that all terms cooccur frequently, it may be able to surface only topics consisting of a smaller number of terms that are therefore more generic / coarse. A question that naturally arises is whether it is possible to formulate a method that lies between the two extremes of a pure FPM approach and typical feature-pivot method that examins only pairwise cooccurrences. Such a method would examine cooccurrence patterns between sets of terms with cardinality larger that two, like FPM does, but it would be less strict by not requiring that *all* terms in these sets cooccur frequently in the same document. Instead, to ensure topic cohesiveness, it would require that large subsets of the terms grouped together, but not necessarily all, cooccur frequently. In the following, we present SFPM, a novel method that attempts to do this. It consists of three

Table 1: Likelihood of appearance of a sample of terms in the new and reference corpuses, as well as the ratio of likelihoods.

Term	New corpus	Ref. corpus	Ratio
Obama	0.03783	0.00004	866.87
Romney	0.03036	0.00004	650.60
Day	0.00175	0.00105	1.65

distinct phases (the algorithm is described in Algorithm 1):

(a) Term selection: The first stage of SFPM involves selecting a set of K terms from the corpus that will be grouped. Any criterion for term selection may be used. We use the approach in [16], which is based on an 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:

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

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). Please note that the sum in the denominator runs through all the terms in the corpus. 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:

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

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. Very common words that may be considered "topic-neutral" are also likely to have a ratio around 1. Table 1 shows the likelihood of appearance of a small number of terms in the reference and the new corpus as well as the corresponding ratio. This illustrates what we already discussed: the names of the candidates are much more likely to appear in the new corpus rather than in the reference corpus. Moreover, a common, "topic-neutral" word like the word "day" is roughly equally likely to appear in both corpora and therefore cannot be considered as important.

(b) Cooccurrence-vector formation: This step is the core of the SFPM method. SFPM 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 already in S. At the end, S will represent a topic. In order to quantify the cooccurrence match between a set S and a term t candidate for inclusion in S, 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. For most cases, apart for very common terms, the D_t will be rather sparse (and similarly D_S will also be rather sparse when not very common terms are included in S), and due to this, an appropriate list-based sparse representation is used. Such a sparse representation also makes the application of the algorithm feasible in relatively large collections of documents.

Considering the definitions of the vectors D_t and D_S, it is clear that 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. 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 the 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 (in the experiments we do not utilize LSH though). A single expansion step of S is displayed in Figure 1.

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 SFPM and is set to be a function of the cardinality of S. In particular, we use a sigmoid function:

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

For appropriate values of b and c, this has the form in Figure 2. The parameters b and c can be used to control the size of the term clusters and how soft the cooccurrence constraints will be. For instance, for the experiments carried out in this paper, b was set to 5 and c was set to 2, resulting in the sigmoid displayed by the blue line in Figure 2. This encourages the addition of terms when the cardinality of S is small (the threshold is low), but makes difficult the addition of terms when the cardinality is larger. A low threshold for the small values of $|S|$ is required so that joining the set S is possible for terms that are associated to different topics and therefore occur in more documents than to ones corresponding to the non-zero elements of D_S. The high threshold for the larger values of $|S|$ is required so that S does not grow without limit.

Additionally, 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 $|S|/2$.

Finally, since we require a set of topics, rather than a single topic, the described greedy search procedure is applied as many times as the number of considered terms, each time initializing S with a candidate term. It should also be noted that it has been observed that if we start from two terms that belong to the same topic, it is not necessary that the two sets produced by the expansion procedure will be the same. Therefore, we initialize S with a different candidate term each time. In some cases though, the produced sets are indeed the same, therefore, we may end up with some duplicate topics.

(c) Post-processing: The final step of the algorithm post-processes the results of the main part by removing duplicate topics.

Regarding the computational complexity of the algorithm, the main factor on which it depends is K, the number of terms that are selected. Considering that the set expansion procedure starts K times, each time initializing the set by a different term, and that in each set expansion step the main operation is to go through the K candidate terms for expansion, the complexity of the algorithm is $\mathrm{O}(K^2)$.

4. EXPERIMENTAL EVALUATION

SFPM was tested on three datasets and compared against a feature pivot algorithm that takes into account only pairwise cooccurrence patterns, a standard FPM algorithm (as described in Subsection 3.1) and LDA. In the following, we first present the three datasets and discuss the evaluation methodology. We then present the feature-pivot method that we experimented with, a graph-based method. Finally, we present the results.

4.1 Datasets

The three evaluation datasets[1] are related to three large real-world events. The first dataset is related to the Super Tuesday primaries, held in the U.S.A in March 2012, the second is related to the FA Cup final, held in May 2012, and the third is related to the U.S.A. Elections held in November 2012. For all three events, a set of tweets was collected using the Twitter streaming API to which a set of relevant keywords was provided. This resulted in a collection of 474,109 tweets for the Super Tuesday dataset; 148,652 tweets for the FA Cup dataset and 1,247,483 tweets for the Elections dataset. Clearly, there is a very large number of topics that appear in these datasets and it would be extremely difficult to manually obtain a set of ground-truth topics manually. Instead, we focus on a specific subset of the topics that appear in the datasets and in particular the ones that have been covered by the mainstream media (in particular, Wall Street Journal, CNN, Guardian, Fox News, Washington Post and Huffington Post). First, a set of topics that were covered in the mainstream media during the period of collection was manually identified. Subsequently, it was verified that these topics are well represented in the collected tweets. In total, 22 topics were identified for the Super Tuesday dataset, 13 for the FA Cup dataset and 64 for the U.S. Elections dataset. Each of the topics is represented by a set of mandatory terms (all of which are required in a candidate topic in order to count it as correctly detected), a set of optional terms (a set of terms which are relevant to the topic but are not required for correct detection of the topic) and a set of forbidden terms (a set of terms that should not be included in the candidate topic, such a set is included to make sure that we can distinguish between closely connected topics). Moreover, the dataset was split into timeslots and the ground-truth topics were allocated to the corresponding timeslots. For some examples of actual topics please see Table 2.

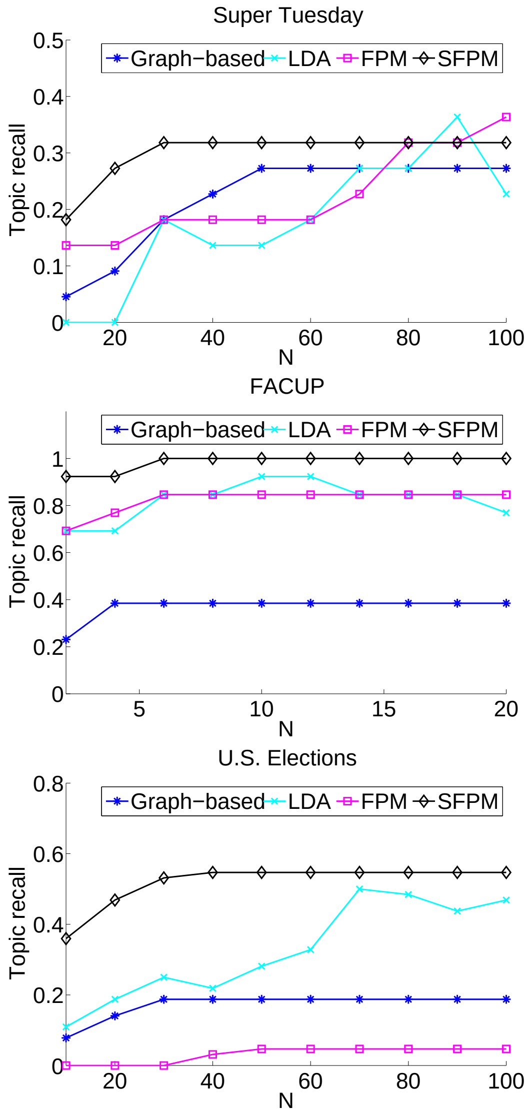

Figure 3: Topic recall for the four tested methods, for the Super Tuesday (top), FA Cup (middle) and U.S. Elections (bottom) datasets.

It should be noted that the identified topics for all datasets are closely related, therefore the datasets and the ground-truth are suitable for testing our assumption that examining cooccurrence patterns of degree larger than two is more suitable than examining only pairwise cooccurrence patterns, in cases where the topics are closely related. To the best of

[1] We made the datasets publicly available at the following location: http://www.socialsensor.eu/results/datasets/72-twitter-tdt-dataset

Table 2: Example ground-truth topics.

Topic	Terms		
	Mandatory	**Optional**	**Forbidden**
Super Tuesday			
Fox reports that Mitt Romney wins Virginia	Mitt, Romney, wins, Virginia	Fox, primary	Ohio, Georgia, Massachusetts, Vermont, Gingrich, Paul, Santorum
NBC reports that Newt Gingrich wins Georgia	Newt, Gingrich, wins, Georgia	NBC, primary	Ohio, Virginia, Massachusetts, Vermont, Romney, Paul, Santorum
Rick Santorum makes a speech about healthcare	Santorum, healthcare	Speech, health, America	Gingrich, Romney, Paul
FA Cup			
Ramires scores for Chelsea	Ramires or Chelsea, score or goal	1-0	
Agger is shown a yellow card for a tackle to Mikel	Agger, booked or yellow card	tackle, Mikel, challenge	
Andy Caroll hits a header but Cech makes a save on the line	Andy or Caroll, line	header, Cech, equalise	
U.S. Elections			
Media report that Mitt Romney wins South Carolina	Mitt or Romney, wins, Carolina	South, CNN, NBC	Barack, Obama
Jesse Jackson is re-elected in Chicago	Jesse, Jackson, re-elected	Chicago, representatives	
CBS reports that Mitt Romney has called Obama to congratulate him	Mitt, Romney, Barack, Obama, call	congratulate, CBS	

our knowledge, there is no publicly available topic detection dataset, accompanied with a ground truth set of topics, that meets this requirement.

The following measures of performance were used:

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

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

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

These scores were computed at the top n topics produced by the topic detection algorithms, for a range of values of n. In order to rank topics, we compute a score for each topic as the average likelihood ratio of its terms.

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* topic that took place at that particular time. A traffic jam and a national election may both be topics appearing in the corpus, and people certainly send tweets about both, but there is no practical way to create a definitive list of all such topics. 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. 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.

4.2 Baseline graph-based algorithm

As a baseline algorithm for our experiments from the class of feature-pivot methods that take into account pairwise cooccurrences between terms, we use a graph-based algorithm. In short, the algorithm steps are the following:

- *Selection*: The top K terms are selected (the same mechanism as for SFPM is used) and a node for each of them is created in the graph G.

- *Linking*: Pairwise similarities between all pairs of terms are computed. Various options are explored (Jaccard, number of doccuments in which terms cooccur, etc.). Nodes of the graph are linked; we experimented with a kNN approach (linking each term with its k nearest neighbours) and an ϵ-based approach (link all pairs of nodes that have similarity higher that ϵ).

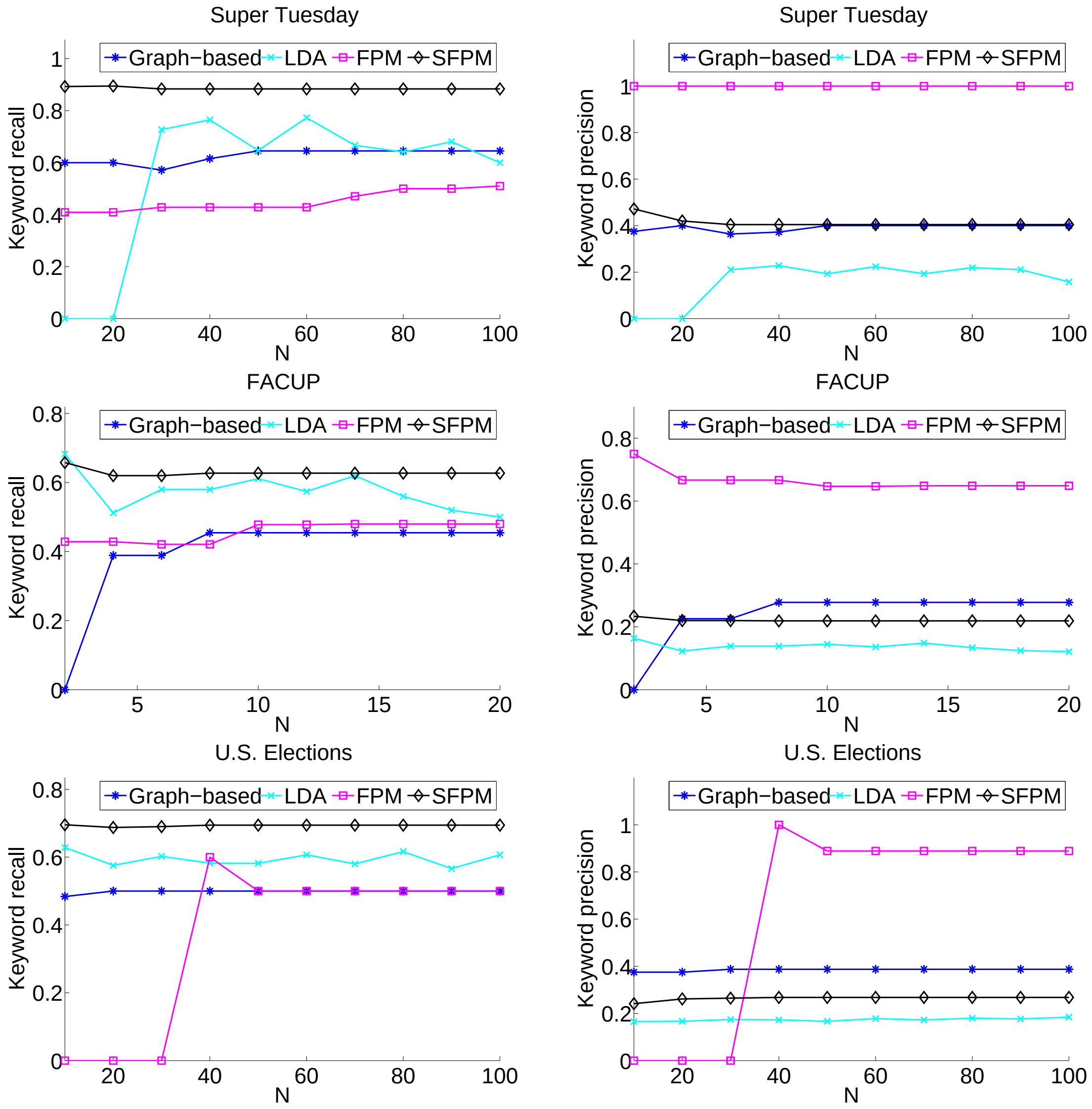

Figure 4: Keyword recall (left column) and keyword precision (right column) for the four tested methods for the Super Tuesday dataset (top row), the FA Cup dataset (middle row) and the U.S. Elections dataset (bottom row). Similar results are obtained for the FA Cup dataset.

- *Clustering*: The Structural Clustering Algorithm for Networks (SCAN) [26] is applied on the graph. Each detected cluster forms a topic. An interesting property of SCAN is that, in addition to detecting clusters of nodes, it provides a list of hubs, each of which may be connected to a set of clusters. The detected hubs may be considered as terms related to more than one topic, something that would not be possible to achieve with a common partitional clustering algorithm. Moreover, SCAN can identify outliers, nodes that are sparsely connected to other nodes; the terms corresponding to such nodes are not eventually linked to any cluster/topic.
- *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 adjacent cluster(s).

Clearly, there are many decisions to be made with respect to the similarity measure used, the construction of the graph, the parameters of SCAN, etc. For this work, the values for these parameters were empirically selected based on a small subset of the available data.

4.3 Results

Table 3 displays the three evaluation measures (@10 for Super Tuesday and U.S. Elections, @2 for FA Cup, due to the smaller number of topics and shorter timeslot duration) for the four tested algorithms. For all datasets, the SFPM approach achieves the highest topic recall. Moreover, it achieves a quite high keyword recall and keyword precision. This indicates that SFPM is able to retrieve more target topics and that it also represents them in a quite complete and accurate manner. FPM performs worse in topic recall but does quite well in keyword precision. This observation is in accordance with the motivation for developping SFPM (cf. Section 3.2): SFPM should be able to detect some of

Table 3: 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.

Super Tuesday

Method	*T-REC@10*	*K-PREC@10*	*K-REC@10*
LDA	0.0000	0.0000	0.0000
Graph-based	0.0455	0.3750	0.6000
FPM	0.1364	**1.0000**	0.4091
SFPM	**0.1818**	0.4717	**0.8929**

FA Cup

Method	*T-REC@2*	*K-PREC@2*	*K-REC@2*
LDA	0.6923	0.6585	0.1578
Graph-based	0.2307	0.4285	0.2857
FPM	0.6923	0.6428	**0.2967**
SFPM	**0.9230**	**0.6666**	0.2186

US Elections

Method	*T-REC@10*	*K-PREC@10*	*K-REC@10*
LDA	0.1094	0.1654	0.6286
Graph-based	0.0781	**0.3750**	0.4839
FPM	0.0000	0.0000	0.0000
SFPM	**0.3594**	0.2412	**0.6953**

the finer granularity topics that FPM cannot, while at the same time we can expect FPM to be more strict in grouping terms than SFPM and therefore a higher keyword precision is to be expected.

Topic recall at a range of values for @N is displayed in Figure 3. Again, for both datasets, SFPM achieves higher topic recall. Interestingly, SFPM produces a smaller number of topics than the other approaches, so the curve is flat after some point. However, topic recall is still higher for SFPM even for higher values of N. For the Super Tuesday dataset though, for large values of @N, it is overcome by FPM and LDA. It is also important to notice that both FPM and SFPM are for most values of @N doing better than the graph-based approach, thereby verifying our assumption. Graphs showing keyword recall and precision for the Super Tuesday dataset are shown in Figure 4. Similar conclusions to the ones obtained by Table 3 can be drawn: SFPM achieves the highest keyword recall and the second highest keyword precision. FPM is doing particularly well in keyword precision but not in keyword recall, meaning that it tends to form smaller but more concrete sets of terms. On the other hand, other methods, including SFPM (due to its “softness” property), produce somewhat less concrete topics. Similar conclusions are obtained by examining the results in the other two datasets.

A sample of topics detected using SFPM can be seen in Table 4.

5. CONCLUSIONS AND FUTURE WORK

This paper presented SFPM, a novel feature-pivot topic detection methods that - as opposed to most existing approaches - examines cooccurrence patterns of degree larger than two. It is based on a greedy set growing algorithm and has been shown to achieve excellent results in three topic detection tasks. Experimental results indicate that SFPM is performing better than competing methods and show a clear improvement over the plain FPM approach. We also argue that FPM approaches such as SFPM can perform better in tasks where the topics are inter-related and fine-grained.

In the future, we intend to test FPM and SFPM with datasets containing different types of documents. Moreover, we intend to examine the problem of synonyms, which is important for short documents like the ones that we experimented with. Finally, we plan to explore other options such as alternative search strategies. For instance, to reduce the computational cost of the method, one could consider stoping the set expansion procedure once the set under consideration matches one of the already identified sets.

Acknowledgments

This work is supported by the SocialSensor FP7 project, partially funded by the EC under contract number 287975.

6. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB ’94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[2] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[3] D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[5] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. *SIGMOD Rec.*, 26(2):255–264, June 1997.

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

[7] C. W. Fox and S. J. Roberts. A tutorial on variational bayesian inference. *Artificial Intelligence Review*, 38(2):85–95, 2011.

[8] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, VLDB ’05, pages 181–192. VLDB Endowment, 2005.

[9] B. Goethals. Frequent set mining. In *The Data Mining and Knowledge Discovery Handbook*, chapter 17, pages 377–397. Springer, 2005.

[10] J. Guo, P. Zhang, JianlongTan, and L. Guo. Mining hot topics from twitter streams. *Procedia Computer Science*, 9(0):2008 – 2011, 2012. Proceedings of the International Conference on Computational Science, (ICCS) 2012.

[11] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions.

Table 4: Sample topics produced using SFPM.

Topic	Terms
Super Tuesday	
CNN Tennessee exit poll: 71% call themselves white, born-again evangelicals	Born-again, Tennessee, CNN, Polls, 71%, Evangelicals
86-yr-old Ohio veteran canŠt vote after government-issued ID is rejected at polling station	86-year-old, Ohio, polling, government-issued
Mitt Romney wins North Dakota	wins, Romney, NBC, Dakota, North, delegates
FA Cup	
Chelsea fans jeered during the national anthem	Scum, sad, jeered, national, anthem, fans, sang
Caroll hits a header but Cech saves	Claiming, Carroll, Cech, sl, saved, header, super, over, liverpool, line
The final ends and Chelsea wins Liverpool with 2-1	Liverpool, 2-1, Chelsea, gone, whistle, sl
U.S. Elections	
Fox projection on results	Fox, News, 123, Tally, Electoral, Obama, 157, Romney, 153, Election
Obama leads on the race for Florida	Barack, Obama, Florida, 50.7, 54.7, Reporting
Twitter activity indicates that Obama will win	Mitt, Romney, Barack, Obama, First, Tweets, decide

Data Min. Knowl. Discov., 15(1):55–86, 2007.

[12] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1):53–87, 2004.

[13] Q. He, K. Chang, and E.-P. Lim. Analyzing feature trajectories for event detection. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 207–214, New York, NY, USA, 2007. ACM.

[14] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[15] H. D. Kim, D. H. Park, Y. Lu, and C. Zhai. Enriching text representation with frequent pattern mining for probabilistic topic modeling. *Proceedings of the American Society for Information Science and Technology*, 49(1):1–10, 2012.

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

[17] J. S. Park, M.-S. Chen, and P. S. Yu. An effective hash based algorithm for mining association rules. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995*, pages 175–186. ACM Press, 1995.

[18] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 181–189, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

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

[20] D. Quercia, H. Askham, and J. Crowcroft. Tweetlda: supervised topic classification and link prediction in twitter. In N. S. Contractor, B. Uzzi, M. W. Macy, and W. Nejdl, editors, *WebSci*, pages 247–250. ACM, 2012.

[21] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

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

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

[24] J. Weng and B.-S. Lee. Event detection in twitter. 2011.

[25] H. Xiao and T. Stibor. Efficient collapsed Gibbs sampling for latent dirichlet allocation. In *Asian Conference on Machine Learning (ACML)*, volume 13 of *JMLR W&CP*, Japan, 2010. (AR: 31

[26] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 824–833, New York, NY, USA, 2007. ACM.

[27] N. Zhong, Y. Li, and S.-T. Wu. Effective pattern discovery for text mining. *IEEE Trans. Knowl. Data Eng.*, 24(1):30–44, 2012.