Maximal Sequence Mining Approach for Topic Detection from Microblog Streams

Fereshteh Jafariakinabad
Department of Computer Science
University of Central Florida
Email: fereshteh.jafari@knights.ucf.edu

Kien A. Hua
Department of Computer Science
University of Central Florida
Email: kienhua@cs.ucf.edu

***Abstract*—Unprecedented expansion of user generated content in recent years demands more attempts of information filtering in order to extract high quality information from the huge amount of available data. In particular, topic detection from microblog streams is the first step toward monitoring and summarizing social data. This task is challenging due to the short and noisy characteristics of microblog content. Moreover, the underlying models need to be able to deal with heterogeneous streams which contain multiple stories evolving simultaneously. In this work, we introduce a frequent pattern mining approach for topic detection from a microblog stream. This approach first uses a Maximal Sequence Mining (MSM) algorithm to extract pattern sequences, each an ordered set of terms. This scheme can capture more semantic information than using unordered sets of the same terms. A pattern graph, which is a directed-graph representation of the mined sequences, can then be constructed. Subsequently, a community detection algorithm is applied on the pattern graph to group the mined patterns into different topic clusters. Experiments on Twitter datasets demonstrate that MSM approach achieves high performance in comparison with the state-of-the-art methods.**

1. Introduction

The term *"microblogging"* was coined in 2006-2007 and since then it has been used to describe social media where users are able to share small units of content. The popularity of online microblogging social media in recent years has led to unprecedented growth in user-generated content, which is a rich source of information about real-world events. The availability of this huge amount of data has initiated research on extracting high quality information by monitoring and analyzing microblogging streams. An example of turning these rich and continuous flow of data into useful knowledge is investigating social media as a sensor to detect real-time events including natural disasters [1]. Moreover, employing it to predict the outcomes of German federal elections [2] offers another example. The main motivation behind all information retrieval frameworks is the real-time reflection of the public's opinion on news as well as their current behavior.

The very first step towards extracting and summarizing useful information from social streams is Topic Detection. Early work on Topic Detection and Tracking (TDT), which was introduced in the late nineties, studied events in news streams [3]. TDT deals with detection and tracking of events from stream of stories. The input stream may or may not be pre-segmented into stories and the events may or may not be known to the system; in other words, the system may or may not be trained to recognize a specific event [3]. Detecting unknown events from streams of stories is more challenging due to lack of any prior knowledge about the event. Even though numerous methods of event detection for conventional news media have been proposed in TDT, the noise of user-based contents and their short length, as well as heterogeneous characteristics of social data streams make it a more challenging task when compared to news streams.

General textual topic detection methods are classified into three classes including document-pivot methods, feature-pivot methods, and probabilistic topic models. In this work, we focus on feature-pivot methods, which cluster terms with respect to their co-occurrence in the corpus. Most of the methods that fall under this category leverage pairwise co-occurrences of terms which yield merged topics in a corpus containing interconnected topics. Frequent Pattern Mining (FPM) is one of the approaches which aims to address this issue by examining simultaneous co-occurrence of more than two terms [4]. Soft Frequent Pattern Mining (SFPM) is a modified version of FPM where a large number of terms must co-occur frequently, but not necessarily all, leading to a soft version of FPM [5]. In this paper, we aim to incorporate relative positional information of terms, as well as distances between terms in a sequence. We argue that this strategy reduces the likelihood of extracting incorrect correlations of terms because the pattern mining is based on term sequence as a pattern which carries more semantic information about the content than an unordered list of terms can. This improvement leads to more accurate topic detection results.

In general, any topic detection method which is based on statistical inferences, is heavily reliant on long documents, while user generated content in social media is usually in the form of short texts. Aggregation is a common solution for addressing this problem in information retrieval. Luca et al. [6] explored the effect of preprocessing steps and the topic detection algorithm itself on social stream.

According to their experiments, in most cases, the time-aggregated datasets achieve lower topic recall scores than non-aggregated datasets. The underlying reason behind this is that the aggregated tweets may represent a mixture of topics rather than a single topic and are therefore more likely to indicate an incorrect association of words. Transaction-level pattern mining, not only results in more informative patterns but also decreases the likelihood of generating mixed topics, since it examines the co-occurrence of terms in the transaction rather than document. Hence, aggregation of tweets does not affect the algorithm in this case. In order to reduce information redundancy in mined patterns we utilize a maximal patterns scheme.

The rest of the paper is organized as follows. We review some related works in the field of topic detection in Section 2. We discuss the frequent pattern mining approaches and introduce the proposed method in Section 3. In Section 4, we present the experimental results that compares MSM against the state-of-the-art methods. Finally, we conclude this paper and discuss potential future works in Section 5.

2. Related Works

General-purpose topic detection methods mainly fall into one of three classes: Feature Pivot Methods, Document Pivot Methods, and Probabilistic Models. Each of these three approaches has advantages and disadvantages. According to Fung. et al. [7], cluster fragmentation problem is one of the common drawbacks of document pivot methods, which leads to incorrect clustering. Probabilistic models usually produce good results; however, they are more computationally expensive. Feature pivot methods, based on analysis of terms correlation, often capture misleading term correlation due to noise in the data set. We discuss these three approaches in more detail in the following subsections.

2.1. Feature-pivot methods

Feature-pivot methods aim to find a group of terms which co-occur in a corpus. In these methods a topic is represented by a set of terms.Generally, feature-pivot methods include two steps towards detection of topics. First, a set of key terms is extracted from the corpus based on some importance measure and then the co-occurrence patterns between these key terms are computed. Second, these patterns are clustered based on some inter-term similarity measures, where each cluster represents a specific topic.

For instance, M. Cataldi et al. [8] consider both term frequency and also social features of tweet, like the popularity of the user, for selection of key terms. They utilize correlation vectors which represent the pairwise co-occurrence of terms in the corpus and generate a graph where each node identifies a term and the edge between two nodes represent the correlation vector of two terms. Finally, a graph-based algorithm is applied for the clustering part.

J. Wang et al. [9] build frequency-based signals for individual terms and detects an event by grouping terms with similar patterns into a set. First, they select bursty terms by filtering away the trivial terms. Then for the clustering part a modularity-based graph partitioning is applied by computing the cross-correlation measures. H. Sayyadi et al. [10] introduced a new event detection method which builds a keyword graph, *"KeyGraph"*, based on the probability of pairwise term co-occurances. The clustering method is a community detection algorithm which iteratively removes the edges with high betweenness. Regardless of the employed techniques for term selection and the clustering part, most of the proposed methods attempt to examine pairwise correlation between terms, while considering correlation of more than two patterns are proposed as follows. J.Guo et al [4] treat the problem of topic detection as a Frequent Pattern Mining problem and propose a stream mining algorithm to detect topics from Twitter streamas. Subsequently, Petkos et al [5] propose a softer version of FPM which represents the topic as frequent patterns.

2.2. Document-pivot methods

Document-pivot methods typically group together individual documents according to their similarity. The similarity measure is computed between either pairs of documents or a document and prototype cluster representation. In this approach a topic is represented by a set of documents. If the similarity of the incoming, document is higher than some threshold, then the document is added to the cluster; otherwise a new cluster is created. The literature works which have adopted this approach mainly differ in the methods they applied to compute the similarity. For instance [11] compares the tf-idf vector of incoming tweets with the tf-idf vector of common terms in each cluster. In [12] a variant of incremental clustering is adopted in which the temporal and textual similarity of incoming tweets are considered. In this method the similarity between incoming news and the clusters older than some limit or those which do not share any textual information is not computed which makes the method more appropriate for large databases. Another document-pivot approach is [13], which aims to address scalability issues by utilizing a modified version of LSH in order to accelerate retrieval of nearest neighbors for each document. In general, the document-pivot methods performance is dependent on the threshold parameter. These methods also suffer from fragmentation issue, for which different merging procedures can be applied [14] [12].

2.3. Probabilistic Models

Probabilistic topic models deal with the distribution of topics and terms. In these approaches, the topic is represented as a distribution over terms. Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Analysis (PLSA) are two representative probabilistic topic models [15] [16] which have been extended widely. They use variables which represent per-topic term distribution and per-document topic distribution. In [17] the supervised version of LDA has been adopted for detecting topics and predicting links in Twitter. H. Kim et al [18] combined the frequent pattern mining method with probabilistic topic models and have reported performance improvements over LDA and PLSA for the classification task.

3. Topic Detection with Frequent Pattern Mining approach

Mining frequent patterns in textual information for topic detection, falls into the class of feature-pivot methods. Early feature-pivot methods in the literature examined the pairwise co-occurrence of terms. This approach suffers from producing mixed topics in heterogeneous streams where several stories are evolving in parallel. One of the solutions for dealing with this challenge is to take into account the co-occurrence of multiple terms rather than just a pair. Needless to say this approach will lead to higher quality results. Idea of exploiting Frequent Pattern Mining for detecting hot topics from twitter was initiated by J. Guo et al. [4], who adopted an FP-stream algorithm in order to discover the patterns in Twitter streams, where a Pattern is a set of terms which co-occur frequently.

G. Petkos et al. [5] proposed SFPM, a soft version of FPM, where a large number of terms in the patterns co-occur frequently rather than necessitating all the terms to appear. It is expected that using SFPM increases incorrectly correlated terms in the mining process, leading to lower keyword precision. L. Aiello et al. [6] compared six different topic detection methods and reported their corresponding keyword precision and recall. Inferred from the reported results, almost all methods have lower keyword precision than recall. This observation implies that most of the terms correlated incorrectly. Hence, we should use a mining algorithm that is able to capture the actual correlation of terms.

The relative positions of terms and the distance between the terms in the corpus can be employed as additional filters for the mining process. In order to capture the relative position of terms in the pattern, we propose to use a frequent sequence mining approach. Sequences are ordered lists of terms that are capable of capturing more semantic information. We adapt the Vertica mining of Maximal Sequential Patterns (VMSP) algorithm and propose a new text mining algorithm which aims to mine maximal sequences. Subsequently, we map the mined sequences into a directed graph, and apply a community detection algorithm in order to cluster the patterns, where each cluster represents a specific topic. In the post-processing step, a set of key terms are selected for each cluster that represents the corresponding topic. In the following, we formulate the task of topic detection from microblogging streams and then describe our approach in details.

3.1. Mining of Maximal sequences

Let text batch B_I be the set of all texts generated by a microbloging stream within a fixed time interval up to the time stamp I. If $T_i(i = 1, 2, \dots, N)$ denotes the topic detected from batch B_I, B_I can be modeled as a set of multiple topics $T_I = \{T_1, T_2, \dots, T_N\}$. The topic detection task in this paper is defined as the task of detecting set T_I from the batch file B_I.

For mining purposes, we consider each user post as an individual transaction. Adopting this approach results in reducing the number of candidates for pattern generation, leading to a lower computational time. Moreover, it ultimately decreases the probability of generating mixed topics since it is unlikely to correlate terms in the different topics.

Definition 0 (Sequence). Let $T = \{t_1, t_2, \dots, t_k\}$ be a set of terms. A sequence $S =< s_1, s_2, \dots, s_n > (s_i \in T)$ is an ordered list of terms. Each user post in the batch is a sequence of terms.

Definition 1 (Sub-sequence). A sequence $\alpha =< a_1, a_2, \dots, a_n >$ is a sub-sequence of another sequence $\beta =< b_1, b_2, \dots, b_m >$, denoted by $\alpha \sqsubseteq \beta$, if there exist integers $1 \leq i_1 < i_2 < \dots < i_n \leq m$, such that $a_1 = b_{i_1}, a_2 = b_{i_2}, \dots, a_n = b_{i_n}$

Definition 2 (Support). Given a batch file $B = \{S_1, S_2, \dots, S_n\}$, where S_i is a sequence representing a transaction in B, $|B|$ is the number of posts in batch B. Let S be a sequence. We call S a sequence of B if there is a $S_i \in B$ such that $S \sqsubseteq S_i$. The support of S is the fraction of posts in batch B that contain S, denoted as $supp(S)$.

Definition 3 (Frequent Sequence). A sequence S is called frequent sequence if $supp(s)$ is greater than or equal to a user-predefined threshold, called the minimum support.

Definition 4 (Length of Sequence). The length of sequence S, denoted as $len(S)$, indicates the number of terms S contains.

In general, a frequent sequence mining algorithm may produce too many patterns which not only makes the task of analyzing the pattern complicated and time consuming but also demands more storage space [19]. A proper pruning scheme can be used in order to reduce the computational cost of the mining task and produce fewer but more representative patterns. Mining closed sequences and mining maximal sequences are two solutions for dealing with inherent redundancy of pattern mining algorithms. Sequences that are not included in any other sequence with the same support are denoted as closed sequences. A maximal sequence is a closed sequence which is not included in another closed sequence. Obviously, output space in the latter is smaller.

Definition 5 (Maximal Sequence). A frequent sequence S is a maximal sequence if there exist no frequent sequence S' such that $S \sqsubseteq S'$.

Term	
Sequence ID	Position in the sequence

Figure 1. Term Vector: An entry of occurrence database

Definition 6 (Occurrence Database). An occurrence database is a database where each entry represents a two dimensional term vector indicating the list of sequences where the term appears along with the position of the term in the sequence. Figure 1 illustrates an entry of the occurrence database.

The initial step towards detecting topics from batch B_I is to find the set of maximal sequences S. Maximal sequence mining is substantial and useful in a wide range of applications; however, few algorithms have been proposed for this task since it is computationally expensive. We adapt the VMSP algorithm in order to discover all the frequent patterns in the batch. VMSP is one of the state-of-the-art algorithms for mining maximal sequences; by adopting a depth-first search method in the database and it is twice as fast as than the previously proposed algorithms [19]. The following text mining algorithm, Maximal Sequence Mining, is proposed to find maximal sequences from a corpus of text:

Algorithm 1 Maximal Sequence Mining

```
Input: B: A batch of tweets
    L: Maximum length of the sequence
    G: The gap between two terms in a sequence.
Output: Set of mined sequences
    Initialization:
    Scan the batch of tweets to create the occurrence
    database and identify $S_{init}$, the list of frequent terms.
    for each term $t$ in $S_{init}$ do
        Find $S_{sequels}$, the set of terms from $S_{init}$ which
        appears after $t$ in batch B.
        return FindMaximalSequence($t, S_{sequels}, L, G$)
    end for
```

Algorithm 2 FindMaximalSequence()

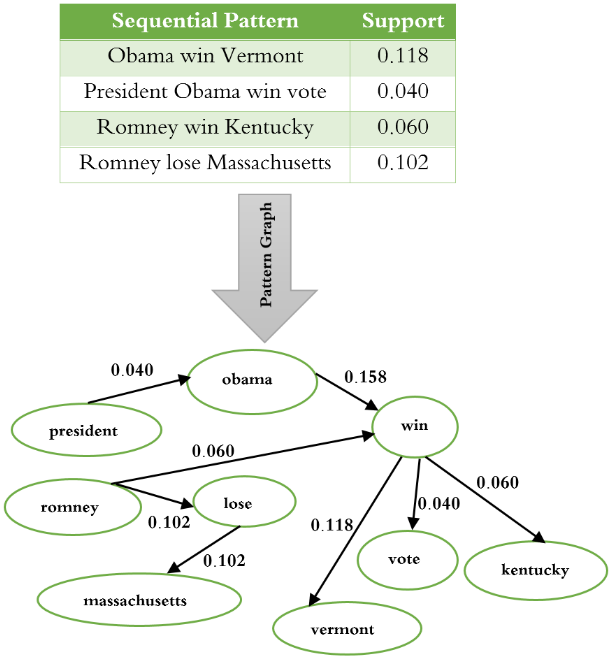
```
for each term $t^{'}$ in $S_{sequels}$ do
    $S_{temp} = \emptyset$, $pattern$ = extension of $t$ with $t^{'}$
    if The extension of $t$ with $t^{'}$ is frequent and the length
    of $pattern$ is less than $L$ then
        $S_{temp} = S_{temp} \cup t$
        $S_{next}$ = Find $S_{sequels}$ the set of terms from $S_{init}$
        which appears after $t$ in batch $B$
        return FindMaximalSequence($pattern, S_{next}, L, G$)
    end if
end for
```

The MSM algorithm is actually finding the longest common subsequences in the corpus and it decreases redundancy of the mined patterns while preventing data loss. However, in most data mining algorithms which adopt Apriori policy long patterns tend not be mined due to the fact that it is less likely to be able to match patterns when the length of the pattern is long [20]. Therefore, long patterns are likely to encounter the low-frequency problem while using static minimum support for all patterns. In order to deal with this challenge we set minimum support very close to 0 which guarantees long patterns with low frequency to be also mined. In order to examine the relative positional information of terms in the topic detection process, we also consider two parameters including maximum pattern length and maximum distance between terms, which serve to control the strictness of the mining procedure.

3.2. Pattern Clustering

Each mined pattern holds some information about a certain topic. In order to generate the final topic, the patterns are clustered into groups where each group corresponds to a specific topic. Initially we map mined patterns into a directed graph, pattern graph, and then apply a community detection algorithm to cluster the patterns into different topics. In what follows, we describe each step in details.

3.2.1. Pattern Graph. A pattern graph is a directed graph in which each node represents a term and the edge between nodes indicates the co-occurrence of terms. Weight of the edge indicates pattern support and the direction implies the order in the pattern. In order to cluster the mined patterns we first, map the mined patterns into the pattern graph. Figure 2 illustrates the graph representation for some instances of mined maximal sequences.



Sequential Pattern	Support
Obama win Vermont	0.118
President Obama win vote	0.040
Romney win Kentucky	0.060
Romney lose Massachusetts	0.102

Figure 2. An example of mapping mined sequences into a pattern graph

3.2.2. Pattern Clustering. We use community detection techniques in order to cluster the pattern graph. Generally, a community in a graph is a subgraph where the nodes are densely connected. Community detection algorithms, sometimes referred as graph partitioning methods, are aimed at dividing vertices of a graph into a number of communities [21]. C. Claudio et al. [22] adopts and edge removal approach for detecting communities, finding the natural divisions of the vertices in a graph without requiring any input parameters, e.g. number of detected communities. The algorithm divides a graph into its subgraphs via iterative removal of the edges based on the edge clustering coefficient C_{ij}. C_{ij} is the fraction of the number of cycles that include a certain edge [22]: and is defined as

$$C_{ij}^{(g)} = \frac{z_{ij}^{(g)} + 1 * A_{ij}}{s_{i,j}^{(g)}}, \tag{1}$$

where $z_{ij}^{(g)}$ is the number of cycles of order g that includes the edge (i, j) with the weight of A_{ij} and $s_{i,j}^{(g)}$ is the number of possible cycles of order g in the given graph [22]. The underlying idea is that the edges between two different communities are unlikely to belong to many short loops. Therefore, inter community edges will have a low value of $C_{ij}^{(g)}$. After removing an inter community edge, the subgraph V is evaluated using the following definition of strong community [22]:

$$k_i^{in} > k_i^{out}, \forall i \in V, \tag{2}$$

where k_i^{in} is the number of edges that connect i to the nodes within V while k_i^{out} is the numebr of edges which connect i to the nodes in the rest of the graph [22]. After applying the algorithm, the graph will be divided into its subgraphs where the vertices are condensely correlated. Then each subgraph presents a set of terms, which co-occur frequently in the corpus.

3.3. Post-Processing

Each cluster generated by the previous step ideally includes all the patterns corresponding to a certain topic. The next step is to extract key terms for topic representation. We define a key node in a graph as a node that has the highest degree. In a graph the degree of a node is defined as follows:

$$D_i = k_i^{in} + k_i^{out} \tag{3}$$

Therefore, each cluster can be represented by a set of key vertices which hold highest amount of degree among all existing vertices in the subgraph. Ultimately, a topic is identified as a set of key terms.

4. Experimental Results

Our method was compared against four other methods including LDA that is a document pivot method, a graph-based method and two frequent pattern mining methods including FPM and SFPM. These methods were tested on three Twitter datasets containing real-world events in different domains. In the following, we first present the datasets and ground-truth data. Then we describe the evaluation method and data preprocessing procedure respectively. Ultimately, we present the experimental results.

4.1. Datasets

The experiments conducted in this paper extract topics from three different datasets of tweets in the sport and political domains which were collected by L. Aiello et al. [6]. The datasets are collections of tweets related to three real-world events in 2012 including FA cup final, U.S.A elections, and Super Tuesday. Each collection is divided into different timeslots and the topics for all timeslots are known. The ground-truth topics include 22, 13, and 64 topics for Super Tuesday, FA cup, and USA Election datasets respectively. These topics are significant topics that are extracted manually and rely on mainstream media reports [6]. It is worth mentioning that the extracted topics are closely related, hence, the proposed datasets and the ground-truths are well-suited for examining the co-occurrence patterns of term. Each topic is represented by the following sets of terms:

- Mandatory terms: these terms must appear in the candidate topic in order to be considered as correctly detected
- Optional terms: these terms may or may not appear in the topic
- Forbidden terms: these terms should not appear in the candidate topic. This set of terms is included in order to distinguish between closely connected topics.

4.2. Data Preprocessing

The preprocessing step plays an important role in the task of topic detection due to the high noisiness of user generated contents, and involves data cleansing and noise removal. We use a preprocessing pipeline that includes the following steps:

- *Tokenization*: This step includes both sentence and word tokenization. A raw tweet is divided into a sequence of terms with hyperlinks, stop words and punctuations removed. Hence, a sequence of cleaner terms is extracted from the raw posts.

- *lemmatization*: In information retrieval, stemming and lemmatization are used to reduce the feature space. Stemming is the task of reducing words to their stem while lemmatization aims to remove inflectional endings in order to return the base or dictionary form of a word, known as lemma. Lemmatization commonly collapses the different inflectional forms of a lemma while stemming most usually disintegrates derivationally related words. In this study, we use lemmatization as it is expected to perform more accurately than stemming.

In order to implement the preprocessing pipeline, we use CoreNLP toolkit [23] to extract clean and noise-free sets of term sequences from the raw datasets of tweets.

4.3. Evaluation

In order to evaluate MSM and compare it against different topic detection methods, we use an evaluation script proposed by L. Aiello et al. [6] where topic recall, keyword precision and keyword recall are the reported evaluation metrics. According to the evaluation method, a topic is correctly detected if it contains all the mandatory terms and none of the forbidden terms. Topic recall is the fraction of ground-truth topics which are correctly detected. The keyword precision is the fraction of correctly detected keywords over the total number of keywords in the candidate topics that have been matched to some ground-truth topics. Keyword recall is the fraction of correctly detected keywords over the total number of keywords in the ground-truth topic that have been matched to some candidate topics. We added F-measure, which is the harmonic mean of keyword precision and keyword recall and it is suitable for measuring overall performance of the methods.

Note that topic precision, which is the fraction of detected topics over the total number of topics that took place at the specific timeslot, was not included in the evaluation. The reason behind is that there is no practical way to produce a definitive list of all topics in the batch, making it impossible to decide if a candidate topic is a real topic that took place in that time interval or no. These measures are computed for the top N topics produced by the detection algorithms. The final performance measure for a dataset is the micro-average of measures corresponding to all timeslots in the dataset. Table 1 shows examples of ground-truth topics and also topics detected using the proposed method.

4.4. Parameter Tuning

In this part, we examine the effect of different parameters on the performance of MSM. Owing to space limitations, we only demonstrate the performance measure results tested on the Super Tuesday dataset. The performance metrics show similar behavior in three different datasets. Figure 3 demonstrates the keyword precision, keyword recall, and keyword F-measure across different values of maximum pattern length (L) and maximum distance between terms (Gap) where the minimum support for mining sequences is set to 0.01(minimum support is explained in Section 3.1). It can be inferred from the charts that precision decreases when increasing L, because longer patterns are more likely to be wrongly correlated terms, causing the detected topics to contain more unrelated terms to the real topics. However, the figure demonstrate keyword-recall grows when increasing the maximum pattern length. The reason is that the longer the mined patterns are, the more information is revealed about the real topics. To see the overall effect of maximum pattern length on the performance of MSM, F-measure is the metric to observe. We can observe from the charts in the figure that F-measure initially grows and then decreases when enlarging L, and generally peaks when L is set to 5. This is expected since F-measure is a trade off between precision and recall.

On the other hand, Gap parameter shows similar behavior. It can be observed from the mentioned figures that all three metrics including precision, recall and F-measure initially increase and then decrease when increasing Gap. A low value of Gap indicates more strictness of the algorithm in grouping terms, causing mined patterns to hold contiguous terms. However, when gap is set to a larger value, MSM will also group terms which are not strongly correlated.

Figure 4 illustrates the topic recall performance of three Twitter datasets across different values of Gap and L. According to the figure, topic recall initially growths when enlarging L and then decreases, and finally tend towards stability when L is enlarged to a certain number e.g. 10. The reason behind is that patterns with the high value of L provide more information about the target topic; however, longer patterns are more probable to wrongly associate terms. Additionally, topic recall decreases when increasing Gap, since the low value of Gap yields to extracting strongly correlated terms. According to the observations from the figures, MSM approach shows its highest topic recall performance when Gap and L are set to 1 and 5 respectively.

4.5. Results

Table 2 shows the evaluation results of topic detection methods for the top N detected topics. For US Elections and Super Tuesday, the top 10 detected topics are considered for evaluations; however, in FA Cup, due to the smaller number of topics and shorter timeslots, the top 2 detected topics are used. Performance evaluation of LDA, Graph-based, FPM and SFPM are reported by G. Petkos et al. [5]. We use the same datasets and same open source evaluation script[1].T-Recall, K-Recall and K-Precision refer to topic recall, keyword recall and keyword precision respectively.

According to the results, MSM approach significantly outperforms the other methods with its best topic recall score for all three datasetes. Moreover, it performs well in keyword precision. This indicates that, as we expected, the MSM approach captures more accurate term correlations due to the use of relative positional information as a filter in the mining process. Although MSM approach performs less in terms of keyword recall, overall it achieves the highest performance in keyword F-measure compared to the other methods in the table. Therefore, MSM approach is able to detect more topics and represent a topic in a more accurate manner. SFPM shows higher performance in keyword recall because, it is not as strict as MSM in grouping terms and clearly correlates more terms. Using more accurate methods

[1] http://www.socialsensor.eu/results/datasets/72-twitter-tdt-dataset

TABLE 1. Topic sample Generated by MSM

Topic	Topic relevant words in ground-truth	Detected Topical Terms
	Super Tuesday	
Mitt Romney wins North Dakota	[mitt romney @mittromney];north;dakota;[win project call lead] cnn;ap	mitt,romney, win, dakota
Rick santorum makes a speech about healthcare	[rick santorum @ricksantorum];healthcare;speech	santorum,speech
	FA Cup	
Agger is shown yellow card for a tackle to Mikel	agger;[booked yellow card];tackle;mikel	mikel, yellow,agger,stoppage,chelsea
The final ends and chelsea wins liverpolll with 2-1	[final whistle gone]; chelsea; champions; congratulations; [2-1 2 1]; win	whistle, go ,chelsea, 2, liverpool ,1 ,final
	US Elections	
Obama wins Wisconsin	[barackobama barack obama]; [win call project held]; [wisconsin wi] cbs; fox	barackobama, win, first ,tweet
Jesse Jackson is re-elected in Chicago	[jesse jackson]; [wins re-elected reelection] ap; chicago; [rep representatives]	barackobama , win

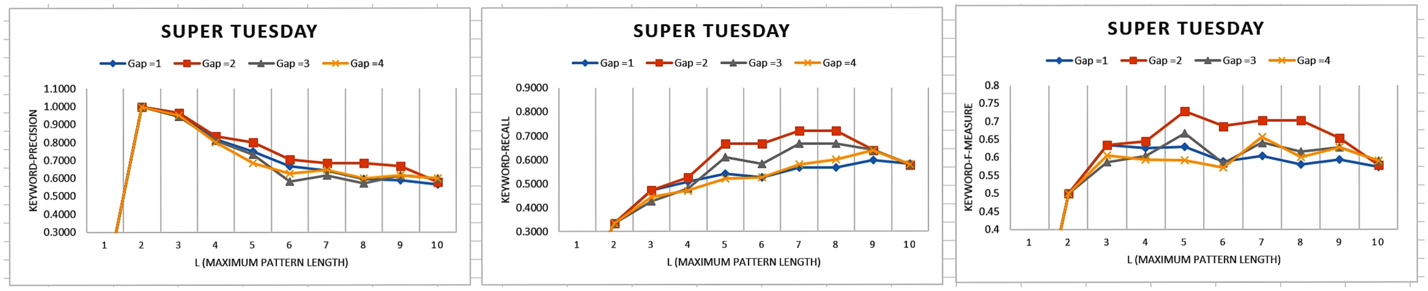


Figure 3. Keyword performance measures across different values of L and Gap in the Super Tuesday dataset

for selecting key terms from a topic cluster may improve the performance of the MSM approach in keyword recall.

TABLE 2. **Performance of topic detection methods in 3 Twitter datasets.**

Super Tuesday

Method	T-Recall	K-Precision	K-Recall	F-Measure
LDA	0.0000	0.0000	0.0000	0.0000
Graph-based	0.0455	0.3750	0.6000	0.4615
FPM	0.1364	1.0000	0.4091	0.5806
SFPM	0.1818	0.4717	**0.8929**	0.6117
MSM	**0.4550**	**0.7500**	0.5410	**0.6285**

US Elections

Method	T-Recall	K-Precision	K-Recall	F-Measure
LDA	0.1094	0.1654	0.6286	0.2618
Graph-based	0.0781	0.3750	0.4839	0.4225
FPM	0.0000	0.0000	0.0000	0.0000
SFPM	0.3594	0.2412	**0.6953**	0.3581
MSM	**0.3910**	**0.6150**	0.5400	**0.5751**

FA Cup

Method	T-Recall	K-Precision	K-Recall	F-Measure
LDA	0.6923	0.6585	0.1578	0.2545
Graph-based	0.2307	0.4285	0.2857	0.3428
FPM	0.6923	0.6428	0.2967	0.4060
SFPM	0.9230	**0.6666**	0.2186	0.3292
MSM	**0.9230**	0.6120	**0.5560**	**0.5826**

5. Conclusion

In this paper, we introduced a Maximum Sequence Mining (MSM) approach, a feature-pivot topic detection method that examines the co-occurrence patterns of terms in the corpus. Its novelty lies in the patterns used for the mining process. They are in terms of sequence of terms as opposed to a set of terms without any particular order. The former pattern representation captures the positional information of the terms in the sequence and is more accurate in reflecting the semantics of the underlying content. Based on this sequence concept, a MSM algorithm is introduced to compute the frequent sequences from a batch of social streams. A directed-graph representation of these sequences, called pattern graph, can then be constructed; and a community detection algorithm is used to partition the nodes in the pattern graph into clusters, each corresponding to a distinct topic. Each topic is represented by a set of keywords selected from the corresponding cluster. Our experiments indicate that the proposed technique performs well in keyword precision. Although it performs less in terms of keyword recall, overall it outperforms current state-of-the-art techniques in topic detection with its best topic recall score. As our future work, we intend to improve the term selection algorithm

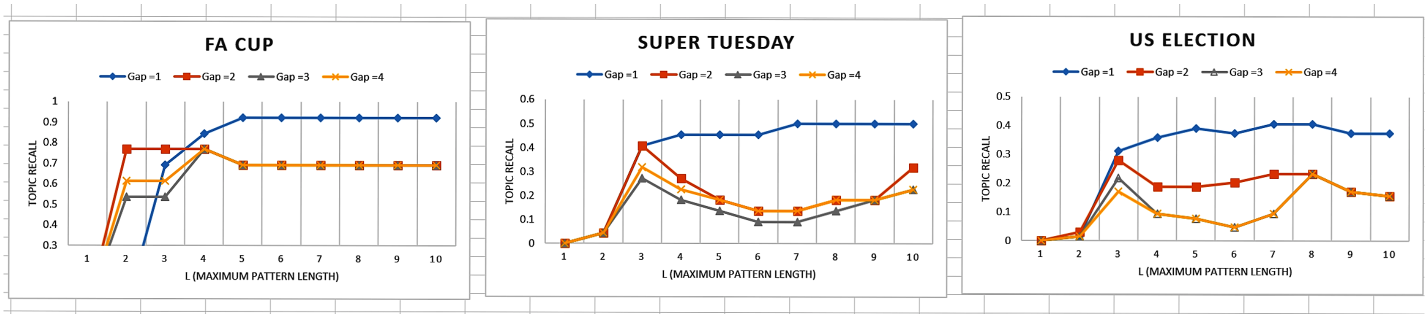


Figure 4. Topic Recall measures of different values of L and Gap for three Twitter datasets

that yield to extracting more representative terms, hence is expected to improve keyword recall. Moreover, we aim to investigate more informative patterns that can hold more semantic information about the target topic.

Acknowledgments

This material is based upon work partially supported by NASA under Grant Number NNX15AV40A. Any opinions, findings, and conclusion or recommendations expressed in this materials are those of the authors and do not necessarily reflect the views of NASA.

References

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

[2] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, “Predicting elections with twitter: What 140 characters reveal about political sentiment.” *ICWSM*, vol. 10, pp. 178–185, 2010.

[3] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang, “Topic detection and tracking pilot study final report,” 1998.

[4] J. Guo, P. Zhang, L. Guo *et al.*, “Mining hot topics from twitter streams,” *Procedia Computer Science*, vol. 9, pp. 2008–2011, 2012.

[5] G. Petkos, S. Papadopoulos, L. Aiello, R. Skraba, and Y. Kompatsiaris, “A soft frequent pattern mining approach for textual topic detection,” in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*. ACM, 2014, p. 25.

[6] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris, and A. Jaimes, “Sensing trending topics in twitter,” *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1268–1282, 2013.

[7] 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 Endowment, 2005, pp. 181–192.

[8] 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*. ACM, 2010, p. 4.

[9] J. Weng and B.-S. Lee, “Event detection in twitter.” *ICWSM*, vol. 11, pp. 401–408, 2011.

[10] H. Sayyadi, M. Hurst, and A. Maykov, “Event detection and tracking in social streams.” in *Icwsm*, 2009.

[11] S. Phuvipadawat and T. Murata, “Breaking news detection and tracking in twitter,” in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2010, pp. 120–123.

[12] 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*. ACM, 2009, pp. 42–51.

[13] 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*. Association for Computational Linguistics, 2010, pp. 181–189.

[14] H. Becker, M. Naaman, and L. Gravano, “Beyond trending topics: Real-world event identification on twitter.” *ICWSM*, vol. 11, pp. 438–441, 2011.

[15] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[17] D. Quercia, H. Askham, and J. Crowcroft, “Tweetlda: supervised topic classification and link prediction in twitter,” in *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, 2012, pp. 247–250.

[18] 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*, vol. 49, no. 1, pp. 1–10, 2012.

[19] P. Fournier-Viger, C.-W. Wu, A. Gomariz, and V. S. Tseng, “Vmsp: Efficient vertical mining of maximal sequential patterns,” in *Canadian Conference on Artificial Intelligence*. Springer, 2014, pp. 83–94.

[20] S.-T. Wu, “Knowledge discovery using pattern taxonomy model in text mining,” 2007.

[21] M. E. Newman, “Detecting community structure in networks,” *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 321–330, 2004.

[22] C. Castellano, F. Cecconi, V. Loreto, D. Parisi, and F. Radicchi, “Self-contained algorithms to detect communities in networks,” *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 311–319, 2004.

[23] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, “The stanford corenlp natural language processing toolkit.” in *ACL (System Demonstrations)*, 2014, pp. 55–60.