

# EVENT DETECTION AND SUMMARIZATION BASED ON SOCIAL NETWORKS AND SEMANTIC QUERY EXPANSION

K. Sathiyamurthy<sup>1</sup> and G. Shanmugavalli<sup>2</sup> and N. Udayalakshmi<sup>3</sup>

<sup>1</sup>Assistant Professor, Department of Computer Science and Engineering,  
Pondicherry Engineering College, Puducherry, India

<sup>2</sup>B.Tech Student, Department of Computer Science and Engineering,  
Pondicherry Engineering College, Puducherry, India

<sup>3</sup>M.Tech Student, Department of Computer Science and Engineering,  
Pondicherry Engineering College, Puducherry, India

## ABSTRACT

*Events can be characterized by a set of descriptive, collocated keywords extracted documents. Intuitively, documents describing the same event will contain similar sets of keywords, and the graph for a document collection will contain clusters individual events. Helping users to understand the event is an acute problem nowadays as the users are struggling to keep up with tremendous amount of information published every day in the Internet. The challenging task is to detect the events from online web resources, it is getting more attentions. The important data source for event detection is a Web search log because the information it contains reflects users' activities and interestingness to various real world events. There are three major issues playing role for event detection from web search logs: effectiveness, efficiency of detected events. We focus on modeling the content of events by their semantic relations with other events and generating structured summarization. Event mining is a useful way to understand computer system behaviors. The focus of recent works on event mining has been shifted to event summarization from discovering frequent patterns. Event summarization provides a comprehensible explanation of the event sequence based on certain aspects.*

## KEYWORDS

*Burst detection, Event detection, Summarization, TLDA, Social networks*

## 1. INTRODUCTION

Event detection and summarization based on social networks and semantic query expansion, this work is proposed for detecting events and to get a summarized output of events based on temporal features. To detect temporal features unsupervised approach for learning is to be modeled in this work. Organization of the events for summarization is based on Hidden Markov Model.

We propose an approach which associates social networks to a given event using query expansion and relationships defined on the Semantic Web, thus increasing the recall whilst maintaining or improving the precision of event detection. A key component of real-time search is the availability of real-time information. Such information has recently proliferated thanks to social media websites like Twitter and Facebook that enable their users to update, comment, and

otherwise communicate continuously with others in their social circle. On Twitter, users compose and send short messages called “tweets”, putting the medium to a wide array of uses. The mining of such social streams is more challenging and different than traditional text streams, because of the presence of both content of text and network structure which is implicitly within the stream. Therefore, the event detection is related to clustering, because the events can only be identified from aggregate changes in the stream.

## 2. RELATED WORKS

A work towards a novel burst-based text representation model for scalable event detection was done [1]. In their work, the author proposed a BurstVSM approach to detect events based on bursty features instead of terms. A work using paraphrases for improving first story detection in news and twitter [2]. In their work, First story detection (FSD) was done to identify first stories about events from a continuous stream of documents. They used paraphrases to alleviate the problem of high degree of lexical variations.

The work towards detecting an event using twitter and structured semantic query expansion was done [3]. In their work they proposed an approach which associates tweets to a given event using query expansion. The results of the work showed the time a topic is tweeted about can be used to identify when an event happened. A detection of events can also follow Tracking in Social Streams [4]. In this paper they proposed the keywords describing an event may be used to find related articles. A work related to detecting events in social tweets gave solution to the two related problems of clustering and event detection in social streams. They studied both the supervised and unsupervised case for the event detection problem [5].

Once the event was detected, it has to be summarized. Therefore the summarization of events using tweets was done [6]. Their work gave solution based on learning the underlying hidden state representation of the event via Hidden Markov Models. Summarization of natural events also possible and work was attempted. In this paper [7] they proposed a novel framework called natural event summarization that summarizes an event sequence using inter-arrival histograms to capture the temporal relationship among events.

A work on summarizing Sporting Events Using Twitter was done based on an automated method for implicitly crowd sourcing summaries of events using only status updates posted to Twitter as a source [8]. The detection of events in web image document stream on social media based on clustering technique integrates with Kleinberg’s burst detection [9]. A work on document clustering with bursty information proposed bursty feature representations that perform better than VSM on various text mining tasks, such as document retrieval, topic modelling and text categorization [10]. For text clustering, proposed a novel framework to generate bursty distance measure. He describes latent Dirichlet allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora [11]. They also present efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes parameter estimation.

Efficient online modelling of latent topic transitions in social media using temporal LDA which gave solution based on TM-LDA is able to highlight interesting variations of common topic transitions, such as the differences in the work-life rhythm of cities, and factors associated with area-specific problems and complaints [12]. An Approach to Semantic Query Expansion introduced a method for semantic ontology searching which combines the approaches of logical reasoning on heuristic inferences and lexical analysis of the user's query to and related concepts

### **3. PROPOSED SYSTEM FRAMEWORK**

#### **3.1. Introduction**

Mining retrospective events from text streams has been an important research topic. There is an efficient work has done separately for event detection and summarization based on temporal features of unsupervised model. There is no combined work for detecting and summarizing an events. So, we proposed a system consisting of both detection and summarization. Therefore, the burst detection algorithm for detecting bursty features was proposed. If an interval of high states appears in the optimal state sequence of some term, this term together with this interval is detected as a bursty feature. In this work we adopted SUMMHMM algorithm for summarization. SUMMHMM takes multiple events of the same type as input, and learns the model parameters  $\theta$  that best fit the data. T-LDA is used to find the temporal features of the events.

#### **3.2. System Architecture**

Our proposed system event detection and summarization is depicted in the figure 1 below. It consists of 5 modules. They are

Data Pre-processing : Cleaning, normalization, transformation, feature extraction of documents includes Data pre-processing.

Burst Detection : To obtain all bursty features in text streams, we perform burst detection on each term in the vocabulary.

TLDA- modeling : The events are listed based on the TLDH- modeling. Summarization: Listed data are summarized.

User query: A search query is a query that a user enters into a search engine for information.

A set of documents are collected from NEWS corpus like ICC, BCI and ESPN news as a dataset related to sports. So we collected sports related reviews from tweeter, facebook etc. We concentrated more on long running events like World cup, IPL and Olympic. These datasets are preprocessed using HTML parsing and segmentation is done for each web documents. These pre-processed output was given as input for detecting the events using Burst detection algorithm. Events which possess higher frequency are detected using this algorithm. These detected events are stored in an event repository.

The keyword given by a user will be searched in event repository. If it is already detected we will proceed with TLDA part. In TLDA the most relevant temporal event data are listed based on occurrence of the event. If not present in event repository we will do the pre-processing and detection methods again. Then these listed data are given for summarization. Then the summarized output is given to the user.

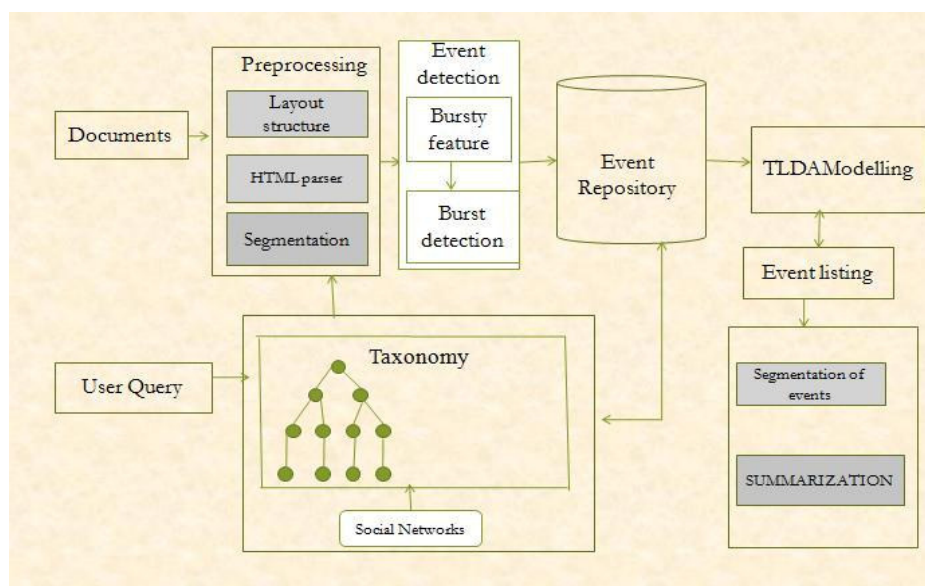


Figure 1. Proposed System Architecture

### 3.2.1 Data Pre-processing

Data pre-processing is an important step in the data mining process. The irrelevant noisy, unreliable data from the collected documents must be removed to avoid the difficulties during training phase. In this work we adopted stanford-pos tagger tool for POS tagging.

### 3.2.2. Burst Detection

To obtain all bursty features in text streams, we can perform burst detection on each term in the vocabulary. The document that obtained from burst detection algorithm is stored in the repository. We have chosen Sports taxonomy as a dataset for topic analysis. It covers large words, topics, keywords and concepts in organized way taxonomy contain section of all the events or keyword related to sports section contains man sub –section. As soon as when the user given the input, the keyword will try to match with taxonomy and based on the event that available in the taxonomy related events are summarized and stored in the repository.

Tweets come in bursts, and the durations of these bursts can vary. If the event is split into constant-time stages, one single long burst can be split into multiple stages, and the key tweets from each stage are likely to be near-duplicates. Conversely, if each stage is too long, it might cover several sub-events in addition to the bursty sub-event; since only a few tweets can selected from each time segment, some sub-events are likely to be missing from the final set of key tweets.

A stream of documents containing a term  $w$  is assumed to be generated from a two-state automaton with a low frequency state  $q_0$  and a high frequency state  $q_1$ . Each state has its own emission rate ( $p_0$  and  $p_1$  respectively), and there is a probability for changing state. If an interval of high states appears in the optimal state sequence of some term, this term together with this interval is detected as a bursty feature.

Given  $B$ , a document  $d_i(T)$  with timestamp  $T$  is represented as a vector of weights in bursty feature dimensions:

$$d_{i(T)} == (d_{i,1}(T), d_{i,2}(T), \dots, d_{i,|E|}(T))$$

We define the  $j$ th weight of as follows

$$\begin{cases} tf - idf_{i,w}^{B_j}, & \text{if } T \in [T_s^{B_j}, T_e^{B_j}], \\ 0, & \text{otherwise.} \end{cases}$$

When the timestamp of  $d_i$  is in the bursty interval of  $B_j$  and contains bursty term  $wB_j$ , we set up the weight using tf-idf method.

### 3.2.3. Split-cluster-merge algorithm for event detection

In this section, we discussed how to cluster documents as events. Since each document can be represented as a burst-based feature vector, we use cosine similarity function to compute document similarities. Here we developed heuristic clustering algorithm for event detection, denoted as split-cluster-merge. It is infeasible to cluster all the documents because of large size of news corpus. The first step is that we split the dataset into small parts, then cluster the documents of each part independently and finally merge similar clusters from two consecutive parts. In our dataset, we find that most events last no more than one month, so we split the dataset into parts by months. After splitting, clustering can run in parallel for different parts, which significantly reduces total time cost. For merge, we merge clusters in consecutive months with an empirical threshold of 0.5. The final clusters are returned as identified events.

### 3.2.4. TLDA Modeling

After preparing and choosing taxonomy of sports as a dataset, the next step is applying a LDA topic analysis model. It can be modeled as a process of generating new documents. The events are listed based on the TLDAH- modeling finally this events are allowed to the Summarization section. A generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. In TLDA, each document may be viewed as a mixture of various topics based on time based events. This is similar to probabilistic latent semantic analysis except that in LDA the topic distribution is assumed to have a Dirichlet prior. T-LDA is meant for getting temporal features of the event. T-LDA learns the transition parameters among topics by minimizing the prediction error on topic distribution in subsequent postings. After training, T-LDA is thus able to accurately predict the expected topic distributions.

### 3.2.5. Summarization

Our goal is to extract a few data that best describe the interesting occurrences in that event. One difference between SUMMHMM and the standard HMM is immediately clear the observation from a given state of SUMMHMM consists of all tweets for that time period (i.e., a multi-set of symbols) instead of just one symbol, as in the standard HMM.

SUMMHMM takes multiple events of the same type as input, and learns the model parameters  $\theta$  that best fit the data. The model parameters consist of multinomial word distributions  $\theta(s)$ ,  $\theta(sg)$ ,  $\theta(bg)$  and the transition probabilities. These parameters are learnt using an EM algorithm. Given  $\emptyset$ , the optimal segmentation of the events can be quickly found by the standard Viterbi algorithm,

which we do not describe here. Each segment can then be summarized, yielding the final set of top tweets for each event. Following Algorithm gives the pseudo-code for our approach.

Algorithm SummHmm

INPUT: Tweet corpus  $Z$  , tweet word vocabulary  $V$  , desired Number of tweets  $n$ , minimum activity threshold  $l$  OUTPUT:Set of key tweets  $T$

Learn  $\emptyset$  by iterating the update equations in (chakrabarti and Punera 2011) until convergence

Infer time segments  $TS$  by the Viterbi algorithm (Rabiner 1989)  $T S' = \{ s \in TS \mid \text{tweet volume in segment } s > l \% \text{ of } |Z| \}$

For each segment  $s \in T S'$  do  $Z [s] = Z$  restricted to time  $s$

$T_s = \text{SUMMALLTEXT} ( Z [s] , V , n / |T S'| )$  End for  
 $T = \cup T_s$

Sentence Score

$$S\_C = \alpha_1 * P\_F + \alpha_2 * T\_F + \alpha_3 * L\_F + \alpha_4 * S\_F + \alpha_5 * C\_F / \sum_{i=1}^5 \alpha_i$$

Where  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  and  $\alpha_5$  are the weights of the position , term , length ,semantic and centrality respectively. These weights are given in order to normalize the values of sentence specific features such that  $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5$  must be 1.

Position Feature:

$$P\_F = \frac{\text{Position of the particular sentence}}{\text{Total sentence}}$$

Term Feature:

$T\_F = \text{Term Weight} * \text{Frequency of the term in the sentence.}$

Length Feature:

$L\_F = \text{Sum of all phrases in the sentence.}$

Semantic Feature:

$S\_F = \text{Position of Keyword in the Topic ontology}$

Sentence Centrality Feature:

$$C\_F = \frac{\text{Keywords in sentence} \cap \text{Keywords in other sentences}}{\text{Keywords in sentence} \cup \text{Keyword in other sentences}}$$

## 4. EXPERIMENT RESULTS

### 4.1. Introduction

This chapter discusses the results of each module in a proposed system. The modules are Data Pre-processing, Burst Detection, TLDA- modeling, Summarization, User query. Each modules will be having outputs and there screenshots are kept as a sample.

### 4.2. Module Results

#### 4.2.1. Data Pre-processing

Data pre-processing is an important and first step in our system. We have already collected the data regarding sport Events. In the database collection all the important events related to sports

and their reviews are collected and stored in the Database. Whenever we want the document to pre-process first we have to load the document which has been chosen from respective path. The reviews which contains the title of the event, date and time of that review, rating for that review and the name of the reviewer.

In the following figure 2 we can see the image of sample dataset of our system. It is named as datas.txt. This is the document which contains all the review about the events and it is allowed for pre-processing. From the figure we can observe that this document contains number of reviews about the events for each event its title, date, reviews, reviewer are tagged separately.

Since we are having the collection of many reviews about the sport events our system will store those event reviews based on their topics. In the table we can see that topics of the events are listed in one column and the reviews related to those topics are listed in separately in another column. For example the events like cricket, volley ball, base ball are listed in one side and it reviews are listed based on the topics.

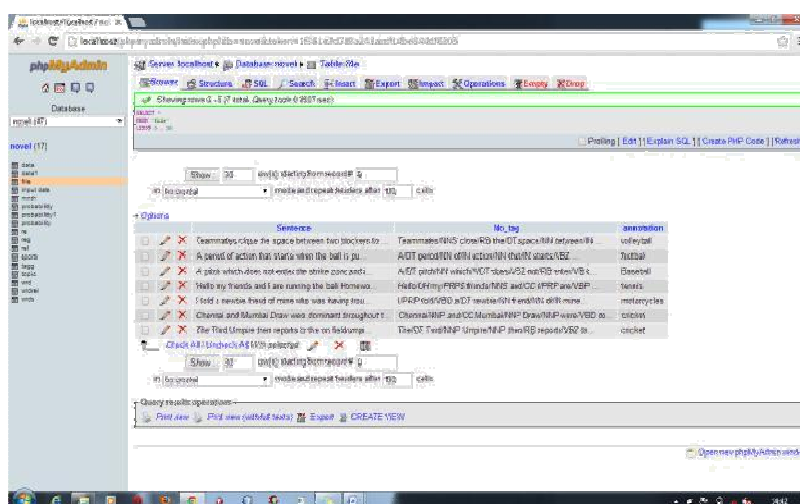


Figure 2. Sample of dataset

For Data pre-processing we have to load the document from its correct path. We have stored the dataset in one separate folder. To pre-process the document we have to select it and open it for pre-processing. The first step in pre-processing is POS tagging. In the part of speech tagging all the words in the document are assigned to the tag. It will tag the words based on their type like verb, noun, Adjective, Adverb. So based on the parts of speech the words are tagged. Based on the tagging of words some particular words are extracted.

From the figure 3 we can see that after the documents are allowed for POS tagging it will be stored in our database table. In our table it contains three columns , the first column contain the actual sentence, The second column which contain the words with their part of speech and the third column which contain the topic of that review. The main aim of POS tagging is to separate all the words based on their speech. We have taken only the nouns and adjective for further processing.



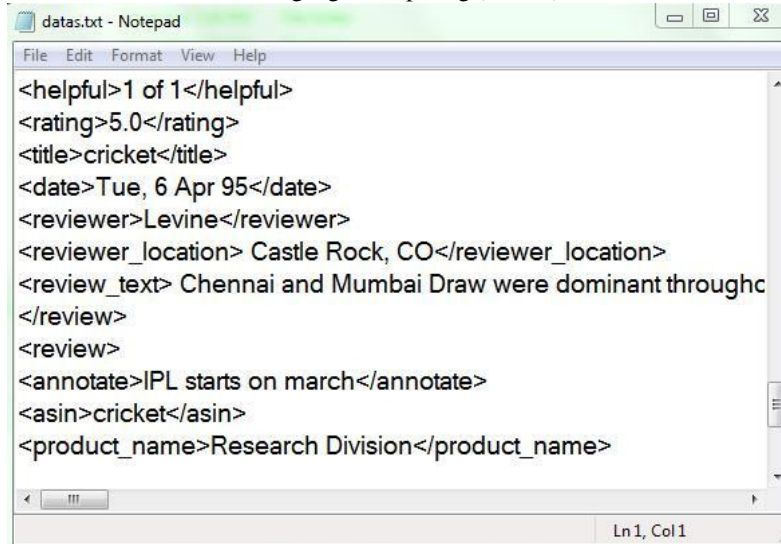


Figure 3. View of Database after POS tagging

#### 4.2.2. Burst Detection

Burst detection is the next module of our system. To find the bursty feature we are using the Burst detection algorithm. Bursty feature is nothing but at some time interval the particular word gets maximum frequency. To find out the frequency of each words we are find out Term frequency , Document frequency , Inverse document frequency and word count of all the words in the document. Through this feature we found bursty feature. Firstly we will find the term frequency of each word in the documents.

TF/IDF is the short form for Term frequency – Inverse document frequency; it is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as weighing factor in information retrieval and Text mining. The tf/idf value increases the probability to the number of times a word appears in the document.

word	C	T	TF	DF	D	IDF
ball	2	12	0.16666667		2	8
4.0	0.041666668					
blockers	1	12	0.083333336		1	1
8	8.0	0.010416667				
game	1	12	0.083333336		1	8
8.0	0.010416667					
opponent	1	12	0.083333336		1	1
8	8.0	0.010416667				
players	2	12	0.16666667		2	8
4.0	0.041666668					
sand	1	12	0.083333336		1	8
8.0	0.010416667					
space	1	12	0.083333336		1	8
8.0	0.010416667					
spiker	1	12	0.083333336		1	8
8.0	0.010416667					

Figure 4. TF, IDF, Document value calculation



From the figure 4 we can see that we have calculated the occurrence of each word in the document that is count of that word is calculated. Based on the count only we have calculated the term frequency. It also shows the values with its count, term frequency, document frequency and its inverse document frequency.

Term frequency is nothing but the number of occurrence of particular word in the document. The Inverse document frequency is the number of occurrence of same word in all the documents.

word	C	T	Probability	Multinomial
ball	2	12	0.16666667	7.168247959180009E-6
blockers	1	12	0.083333336	3.5841239795900046E-6
game	1	12	0.083333336	3.5841239795900046E-6
opponent	1	12	0.083333336	3.5841239795900046E-6
players	2	12	0.16666667	7.168247959180009E-6
sand	1	12	0.083333336	3.5841239795900046E-6
space	1	12	0.083333336	3.5841239795900046E-6
spiker	1	12	0.083333336	3.5841239795900046E-6
team	1	12	0.083333336	3.5841239795900046E-6
teammates	1	12	0.083333336	3.5841239795900046E-6

word	C	T	Probability	Multinomial
act	1	19	0.05263158	2.064188176577341E-9
action	1	19	0.05263158	2.064188176577341E-9
arm	1	19	0.05263158	2.064188176577341E-9
ball	3	19	0.15789473	6.192564237523419E-9

Figure 5. Probability and multinomial value

Finally we have to calculate the probability. The probability of each word is nothing but occurrence of each word probability. So that we can find which word is having maximum frequency that can be consider as bursty feature.

### 4.2.3. User Query

There is typically not a single page that contains all the information sought; indeed, users with event keyword typically try to assimilate information from multiple pages. A search keyword is a event keyword that a user enters into a search page to satisfy information needs. When a user gives a keyword, he will get a summarized output of the detected event data.

### 4.3 Performance Metrics

Precision is the fraction of generated events that are relevant to us, while recall (also known as sensitivity) is the fraction of relevant events that are generated by the system. In simple terms, high recall means that an algorithm returned most of the relevant results, while high precision means that an algorithm returned substantially more relevant results than irrelevant.

$$\text{Precision} = \frac{\text{Relevant events generated by system}}{\text{Total events generated}}$$

$$\text{Recall} = \frac{\text{Relevant events generated by system}}{\text{Total events to be generated}}$$

Perplexity:

TM-LDA is adopted to predict the topic distribution of future Tweets based on historical Tweets. The measurement of Perplexity to evaluate TM-LDA against the actual word occurrences in future Tweets. Perplexity is used to measure how well a language model fits the word distribution of a corpus. It is defined as:

$$\text{Perplexity} = 2^{-\sum_{i=1}^N \log_2 p_l(x_i)}$$

The above formula depicts the perplexity of the language model l, where  $p_l(x_i)$  is the probability of the occurrence of word  $x_i$  estimated by the language model l and N is the number of words in the document. The language model yields higher probability for the occurrences of words in the document than words that are not in the document, the language model is more accurate and the perplexity will be lower.

Table 1. Results of evaluation measures for Existing system and proposed system

Metrics	Existing System	Proposed System
Precision	80	84
Recall	74	84
Perplexity	20	17

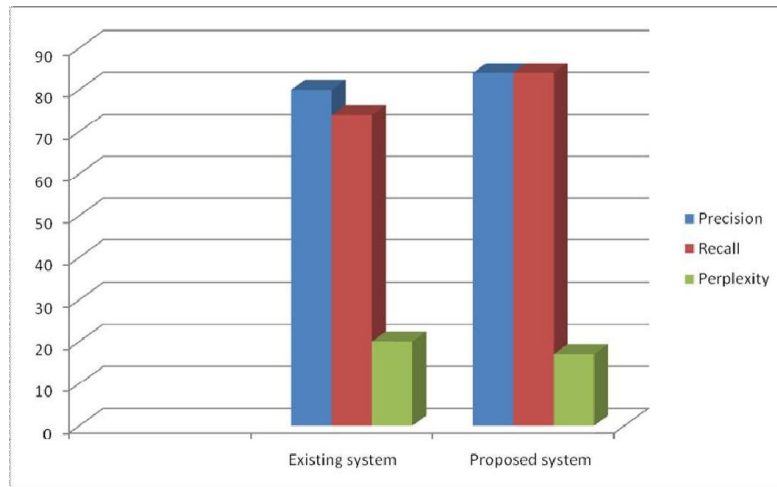


Figure 6. Resulting Graph of Existing system and Proposed System

The above resulting graph depicts the existing system performance with proposed system . Compare to existing system the proposed system shows higher precision and recall and relatively low perplexity. We have shown that our method is able to more faithfully model the word distribution of a large collection of real micro-blogging messages, compared to previous state-of-the-art methods. Furthermore, we introduced an efficient model updating algorithm for TM-LDA that dramatically reduces the training time needed to update the model, making our method appropriate for online operation. In a series of experiments, we demonstrated ways in which TM-LDA can be naturally applied for mining, analyzing, and exploring temporal patterns in micro

blogging data.

## 5. CONCLUSION

Classic text representation model (i.e., vector space model) cannot model temporal aspects of documents. Twitter has become more popular for satisfying user queries. Recent research has shows fraction of tweets are about “events”, and the detection of novel events in the tweet- are attracted alot. Latent topic analysis has emerged as one of the most effective methods for classifying, clustering and retrieving textual data. In contrast, textual content in the web has especially in social media, is sequenced temporally, includes microblog posts on sites such as Twitter and Weibo, status updates on social networking sites such as Facebook and LinkedIn, or comments on content sharing sites such as YouTube. Event detection based on Temporal modeling enables to increase the accuracy for the user query. Event summarization based on temporal and Hidden Markov Model enables achieve effective ordering of events for summarization.

In this work different approach for temporal information retrieval are corresponds to dimensions of bursty features inplace of terms, which can retrieves semantic and temporal information. We formalize the problem of summarizing event-tweets to give a solution based on the underlying hidden state representation of the event via Hidden Markov Models. We presented and evaluated a temporally-aware language model, TM-LDA, for efficiently modelling the topics and topic transitions that naturally arise in document streams.

## 6. FUTURE ENHANCEMENT

In this work of Event detection and summarization of temporal data is attempted for knowledge and Language modeling of LDA. This work can be explored to further levels of more huge domains. Event Detection in our project is based on the matching of set of keywords which can be extended by the using ontology. This work can also be applied to all other topics as well.

## REFERENCES

- [1] Rishan Chen, Kai Fan, Hongfei Yan & Xiaoming Li, (2012) "A novel burst-based text representation model for scalable event detection", in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Vol.2, pp43-47.
- [2] Saša Petrović, Miles Osborne & Victor Lavrenko, (2012) "Using paraphrases for improving first story detection in news and twitter", in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp338-346.
- [3] Heather S. Packer, Sina Samangooei, Jonathon S. Hare, Nicholas Gibbins & Paul H. Lewis, (2012) "Event detection using twitter and structured semantic query expansion", in Proceedings of the 1st international workshop on Multimodal crowd sensing, pp7-14, ACM New York, NY, USA.
- [4] H. Sayyadi, M. Hurst, & A. Maykov, (2009) "Event Detection in Social Streams", in Proceedings of Third International AAI Conference on Weblogs and Social Media.
- [5] Charu C. Aggarwal & Karthik Subbian, (2012) "Event Detection in Social Streams", in Proceedings of the Twelfth SIAM International Conference on Data Mining.
- [6] Deepayan Chakrabarti & Kunal Punera, (2011) "Event summarization using tweets", in Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011.
- [7] Yexi Jiang & Chang-Shing Perng, (2011) "Natural Event Summarization", in Proceedings of the 20th ACM international conference on Information and knowledge management, pp765-774.

- [8] Jeffrey Nichols, Jalal Mahmud & Clemens Drews, (2012) "Summarizing Sporting Events Using Twitter", in Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, pp189-198.
- [9] S.Tamura, K.Tamura, H.Kitakami & Hirahara, (2012) "Clustering-based Burst-detection Algorithm for Web-image Document Stream on Social Media", in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp703 - 708.
- [10] Apirak Hoonlor, Bolesław K. Szymanski, Mohammed J. Zaki & Vineet Chaoji, (2012) "Document clustering with bursty information", in Proceedings of Computing and Informatics, pp1533-1555.
- [11] D. M. Blei, A. Y. Ng and M. I. Jordan,(2003) "Latent dirichlet allocation", J. Mach. Learn. Res., vol. 3, March, pp. 993–1022.
- [12] Yu Wang, Eugene Agichtein & Michele Benzi, (2012) "TM-LDA: Efficient online modelling of latent topic transitions in social media", in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp123-131.
- [13] J Malek & V Rozinajová, (2003) "An Approach to Semantic Query Expansion", in Proceedings of 26th Information Systems Research Seminar in Scandinavia, August 9-12, Porvoo, Finland.