Users' Topic Detection from Tweets based on Keyword Extraction

G. Hemantha Kumar
Dept. of Computer Science,
University of Mysore,
Mysore, India

Seyedmahmoud Talebi
Dept. Of Computer Science,
University of Mysore,
Mysore, India

Manoj K.
Bengaluru,
India

ABSTRACT
In this paper, a different approach for detecting users' topic of interest in twitter based on keyword extraction methods and neural network has been shown. An approach to Text Mining is proposed by extracting the topics relevant to some keywords and further used in predicting topics from users' tweets (Twitter posts). The TF-IDF method has been used to extract keywords in this work. The proposed method, which uses neural network, has been shown to be efficient for topic detection and further comparison. Back propagation method is used to train and to learn the neural network.

General Terms
Social network, Text Mining, Knowledge Base System, Natural Language Processing. Information Science.

Keywords
Neural Network, Keyword Extraction, Topic Detection, Twitter.

1. INTRODUCTION
Social network nowadays produces an environment in which users could spend a lot of time on it and use it for different purposes. Based on this interaction between users, we have a huge amount of data for each individual user. For analyzing purposes, we could have a list of interested topics for each user on a social network. Therefore, detecting a user's topic, in which the person is interested in, could be immensely helpful. For this purpose, we need to use some text mining methods.

Text mining is a new area in which extracted information from unstructured data is useful. Several approaches exist for identification of patterns, including dimensionality reduction, automated classification and clustering [1]. Figure 1 shows the process steps for text mining [2].

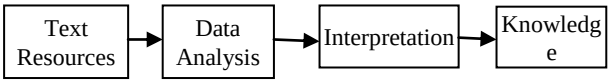

Fig 1: Text mining process

Topic detection is the task that gives very brief and efficient words which maybe unknown to the system. Topic is the abstraction of the whole document or some part of the story. There are different methods for approaching topics like Vector Space Model, Hierarchical Clustering, Named Entity Recognition Model, Hidden Markov Model, and Based on Keyword Extraction System etc. [3].

According to Fiscus and Doddington[4], the basic method for defining TDT are:

- Topic Tracking – detect stories that discuss a target topic,
- Link Detection – detect whether a pair of stories discuss the same topic,
- Topic Detection – detect clusters of stories that discuss the same topic,
- First Story Detection – detect the first story that discusses a topic, and
- Story Segmentation – detect story boundaries.

2. LITERATURE SURVEY
Topic Detection and tracking is the area in Information Retrieval. Began during 1996 and 1997 to explore various approaches and establish performance baseline. The research began in 1996 with DARPA. In recent years, TDT techniques have been developed to identify the issues discussed in a large collection text [5]. Different techniques have been designed such as [6]:

1) Based on Hidden Markov Models [7] [8]: This approach makes use of hidden Markov modelling and clustering techniques. A stream of unsegmented text (as might be generated from automatic transcription of broadcast news, for example) is regarded as being composed of a series of "topics" in something like the same way that a stream of speech consists of a series of phonemes. A story on a particular topic can then be viewed as analogous to an utterance of a particular phoneme, and a stream of text can be decoded into a series of topics in the same way that a speech recognizer decodes a stream of speech into a series of phonemes. The boundaries of these topics are identified with story boundaries.

2) Probabilistic Model [9]: Topic detection based on probabilistic models which has been used for news and media.

3) Based on NER: Wang Xiaowei, JiangLongbin, MaJialin and Jiangyan [7] came up with a new improved approach for topic tracking. Due to the high dimensionality of Vector Space Model, some important characteristics of the text are usually submerged by many weak ability characteristics. They proposed multi vector model that extracts NER features from text and make it into a separate vector. It first selects the features and classify in accordance with characteristics of different tasks, then calculates the vector, then finally selects the combination of model and optimize the parameters. Their experimental result shows that the tracking performance is improved by using multi vector model.

4) Based on Statistical Language Modeling: Maximilian, Michal, Cai-Nicolas and Dietmar [10] proposed an approach which allows monitoring news wire on different levels of temporal granularity, extracting key

phrases that reflect short term topics as well as longer term trends by means of statistical language modeling. The focus is on observing the development of topics over time. Modeling these, developments over time lend tools to track and analyze topics in a method independent of time slices by themselves. The approach uses several tiers of sliding windows in order to capture topics of varying longevity. Representative keyword vectors are established by discovering salient terms within a topic, such that common topic terms are uncommon within the text corpus in general.

3. ALGORITHM DESIGN

Topic detection based on keyword extraction will be performed in four general steps as below:

Step1: Remove Repeated Posts and stop words from each posts.

Step2: Extracting keyword of the user tweets based on TF-IDF method TF-IDF value is composed of two components TF and IDF values. The rationale of TF value is that more frequent words in a document are more important than less frequent words. TF value in a document is the number of times a given term appears in that document. The second component of TF-IDF value is, IDF, represents rarity across the whole collection. This value is obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$tf(i,j) = \frac{n(i,j)}{\sum_k n(k,j)} \quad \text{-i}$$

$n(i,j)$: The number of occurrences of the considered term in document d_j

$\sum_k n(k,j)$: The number of occurrences of all term in document d_j

$$idf(i) = log\left(\frac{|D|}{|dj:tj \in dj|}\right) \quad \text{-ii}$$

$|D|$: The total number of documents in the corpus

$|dj:tj \in dj|$: Number of documents where the term t_i appears

$$tfidf(i,j) = tf(i,j) \times idf(i) \quad \text{-iii}$$

***Step3*:** Cluster posts based on their similarities.

Step4: Keywords of each cluster would be the input of NN trained network to get list of related topics.

4. ANALYSIS AND EVALUATION

For this work, we have list of users' tweets (we can access to users' tweets based on some API) which are available in different platform. The one that we used was tweepy API, which is in python language and it provides us a timeline of users as well as more data, such as list of followers, etc. Let us consider each tweet as a document. Topic detection from the whole number of documents needs some pre-processing initially. At the first step, we will remove stop words and repeated posts. Stop words SW are such as ["at"," the"," how" etc.].

We now have list of cleaned twitter posts or we have list of cleaned documents. Tweets = [list of tweets of all users]. Each tweet has a list of posts. Therefore, for each user/tweet we are removing repeated words and SW. This new list of tweets will be the input of keyword extraction algorithm to extract keywords. After that with cosine similarity, we cluster keywords. It means that we cluster posts, which are similar to each other. Now, new tweets will be the input of a neural network, which has been trained to work as an unsupervised topic detection of this work. The network will give the output, which is the index of the lookup table, which gives the list of topics. For this neural network we use back propagation method because we need method to create network, which could update based on the input and finally are better trained. In Figure 2, we are presenting the general overview of the algorithm and the process for each of the steps. After training the network, we call it Topic Detector Network (TDN) we use it to test for the inputs from users' tweets. The detail of each step will be as following.

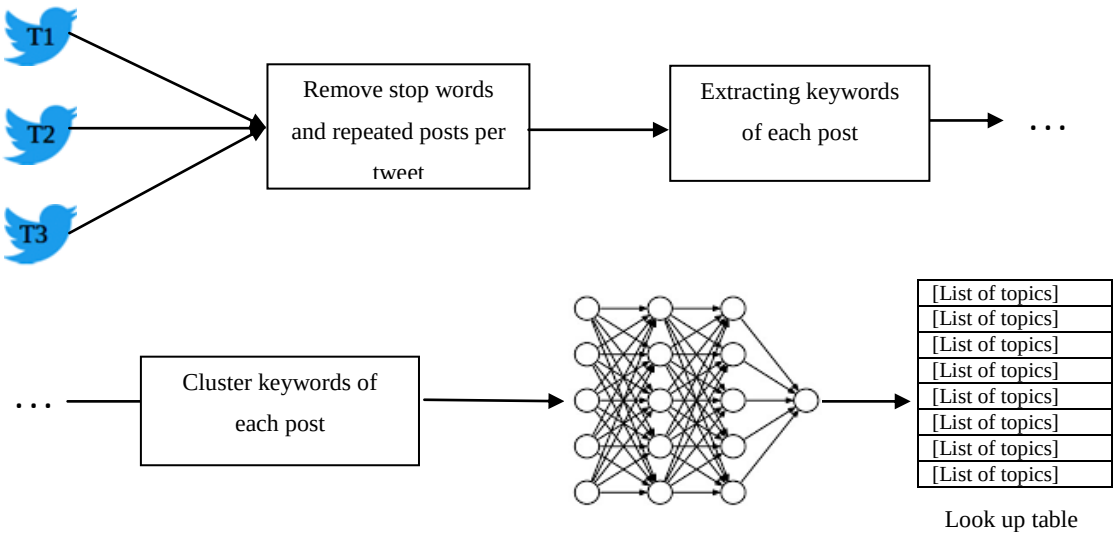

Fig 2: General overview of the whole processes

From Step 1 of the proposed algorithm design, removing those unwanted part of the text is a search for each word to check whether that word is an unwanted word or not, therefore a O(n) time complexity we have for this step. At Step 2, for each post of tweets we extract keywords. Keywords extraction use TF-IDF algorithm which is taking O(|T|.|D|) time complexity where D is number of document and T is number of terms. At Step 3, we calculate the similarity between posts of each tweet to group them in same cluster. We use cosine similarity to calculate distance of each pair of posts and cluster them by the distance. Therefore, top 10 most similar to each post will be in same group. Time complexity for cosine similarity is O(|W|) which W is number of common words in both input posts as an input of cosine function. This calculation is between each post with whole therefore we would have $O(n^2)$ time complexity for this step.

At this point, we had grouped the posts, which would be the input of neural network of the next step of this work. The network has been trained based on the BBC news and most applicable topics and keywords which using nowadays. We consider news from BBC with most coverage of all topics, which is common nowadays. The reason we refer to BBC is that it covers any aspect of people's life whether technologically, politically and etc. for each news BBC present some topics, which is 3-4 key-phrases, and each key-phrase consist of 2-3 words. The keywords of each content of the news and topics that are related to that news will be the input for NN for training. After training the network, we use in our method, which based on the input for the network and will give the index of the lookup table, which has the list of topics. Figure 3 presents the creation of the mentioned network.

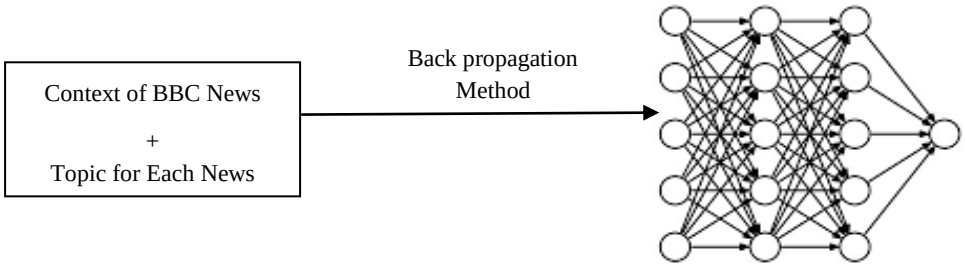

Fig 3: Presenting of creating TDN

After explanation of the methods, we can check the performance of these methods. We are predicting the time by analyzing the algorithm as shown below in Figure 4, Figure 5 and Figure 6, for a hardware system with Intel Core i5 3.0 GHz CPU and 16GB DDR3 Ram. Based on this hardware configuration our prediction for unit of the time will be in the order of Nano seconds.

At Step 1, for all number of terms in tweets we should check whether it is stop word or not and remove repeated posts, as shown in Figure 4.

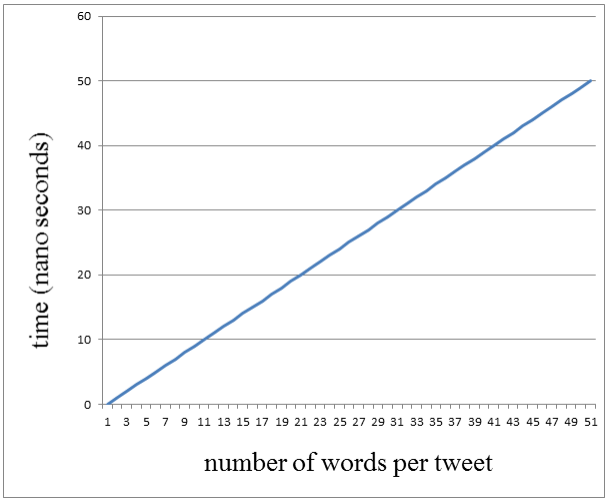

Fig 4: Time analysis of Step1

At Step 2, we have n number of terms in each tweet and m number of tweets which we can consider $O(n^2)$ complexity, as shown in Figure 5.

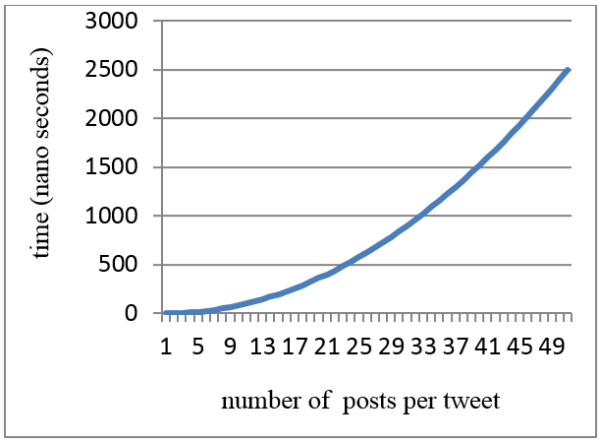

Fig 5: Time analysis of Step2

Moreover, at Step 3, similarity of each post is calculated with others and then clustering them based on distance. Therefore, as it mentioned $O(n^2)$ will be complexity of that as shown in Figure 6.

The time complexity shows that, except the training, which is the initiate step of this idea; the remaining is very optimized and consumes less amount of time. Compare to other methods it does not take too much for processing except for the pre-processing steps, which is minimal.

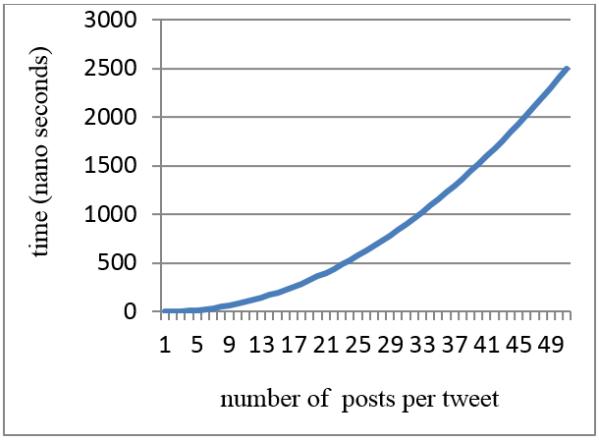

Fig 6: Time analysis of Step3

5. CONCLUSION

The proposed work has two main contributions. After extracting user's posts through API, we have proposed a fast neural network based method for extracting appropriate topics based on specific relevant keywords. It is then shown that by creating clusters based on keywords would further help in easier detection of topics from users' tweets. The methods, which were presented, are known to be reliable and fast.

6. REFERENCES

[1] M. Radovanović and M. Ivanović, "Text mining: Approaches and applications," Novi Sad J. Math, vol. 38, no. 3, pp. 227–234, 2008.

[2] A. Stavrianou, P. Andritsos, and N. Nicoloyannis, "Overview and semantic issues of text mining," ACM Sigmod Record, vol. 36, no. 3, pp. 23–34, 2007.

[3] K. Kaur and V. Gupta, "Topic tracking for Punjabi language," Computer Science & Engineering: An International Journal (CSEIJ), vol. 1, no. 3, pp. 37–49, 2011.

[4] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," in Topic detection and tracking, Springer, 2002, pp. 17–31.

[5] A. K. Kolya, A. Ekbal, and S. Bandyopadhyay, "A simple approach for Monolingual Event Tracking system in Bengali," in Natural Language Processing, 2009. SNLP'09. Eighth International Symposium on, 2009, pp. 48–53.

[6] K. Kaur and V. Gupta, "A survey of topic tracking techniques," International Journal 2, vol. 5, 2012.

[7] W. Xiaowei, J. Longbin, M. Jialin, and others, 'Use of NER Information for Improved Topic Tracking', in Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on, 2008, vol. 3, pp. 165–170.

[8] J. P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt, "A hidden Markov model approach to text segmentation and event tracking," in Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, 1998, vol. 1, pp. 333–336.

[9] F. Walls, H. Jin, S. Sista, and R. Schwartz, "Probabilistic models for topic detection and tracking," in Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, 1999, vol. 1, pp. 521–524.

[10] M. Viermetz, M. Skubacz, C.-N. Ziegler, and D. Seipel, "Tracking topic evolution in news environments," in E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on, 2008, pp. 215–220.