Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2017, 4 (5): 383-389

Research Article ISSN: 2394 - 658X

Recommendation of Hashtags for Effective Organization of Tweets Using Frequent Pattern Mining

Pooja GR[1], Sameeksha M[2] and Asha T[1]

[1]*Department of Computer Science and Engineering*, Bangalore Institute of Technology, *Bangalore, India*
[2]*Department of Computer Science and Engineering*, Kammavari Sangha Institute of Technology, *Bangalore, India*
poojaguptahsn@gmail.com

ABSTRACT

Twitter has evolved into a powerful communication and information sharing tool used by millions of people around the world and is currently overwhelmed by massive amount of tweets generated by its users. To categorize their tweets many users, use hashtag. A hash tag, a keyword prefixed with a hash symbol (#), is a feature in Twitter to organize tweets and facilitate effective search among a massive volume of data. However, hashtag is not restricted in any way in terms of usage, which leads to large number of hashtags in twitter data set. Furthermore, hash tags can be used to collect user opinion on public events, community etc. However, there are very few tweets containing hashtags, which impedes the quality of search results and their further usage in various applications. Therefore, hashtag recommendation has become a particularly important research problem. Existing methods have proposed a personalized hash tag, content based hash tag and hash tag recommendation using TF-IDF approaches which is quite complex. In this paper we address an automatic hashtag recommendation system which uses simple map reduce functions and rule mining techniques. Map reducing function takes the tweet as input and process the given tweet to give key value pair as output. This Output is fed back to rule mining techniques to generate a frequent itemset and that is recommend as hash tag. Our theoretical approach says that this method recommends a hashtag which is more stable and reliable than other approaches.

Keywords: Twitter, Hashtag, TF-IDF, Map Reduce, Rule Mining

INTRODUCTION

Social networks have gained significant importance on the web from many years. The most popular microblogging tool, Twitter has experienced tremendous success lately and has become very important as both a social network and a news media. Users can post text messages of up to 140 characters to tell others 'what they are doing' or 'what is happening' [3]. User-generated short messages are called tweets. By following others, users can keep up with their latest posts. Tweets can contain URLs, embedded images/videos, user mentions, locations, and hashtags. We will be focusing on the hashtag recommendation in this paper. Hashtags are words prefixed with '#' and are used to indicate the topics of tweets. For example, '#BIT 2017' can be used in tweets related to BIT events or announcements. By clicking hashtags in Tweets, users can view all Tweets containing the hashtag. Extremely popular hashtags often become trends. In fig 1.1 how hashtag used in tweet is shown. However, the use of hashtags is still not widespread on Twitter as only 20% of tweets contain hash tags [5]. This can be mainly attributed to the fact that assigning hash tags to tweets is cumbersome and time consuming. Nonetheless, hashtags are useful for categorization and discovery of content and conversations in online social networks.

Functions of Hashtag

Hashtags plays an important role in Twitter. Popular hashtags can become trending topics in the home page of Twitter. The functions of hashtags are briefly summarized as follows: (1) Users can categorize and search tweets by hashtags. (2) Hashtags can lead to temporary discussion groups driven by special events or interests. (3) Hashtags are the core elements in event detection and tracking [2], tweets retrieval [9], analysis of information diffusion [9], and advertising [1]. Thus annotating tweets with the right hashtags is the foundation for many high-level applications. Example for Tweets with hashtag is shown in fig 1[13]

Fig. 1 Hashtag [13]

Problems Faced in Hashtag Recommendation

Despite the great importance of hashtags, a few problems remain unsolved when a user wants to annotate a tweet:

- Before creating a new hashtag, is there any way for the user to find out whether some related hashtags have already been created and widely used? Without hashtag recommendation, hashtags can easily explode since different users may choose different words as hashtags to describe the ongoing topic, although some of them may represent similar meanings. On the other hand, users can only handle a portion of information they receive [1]. Hashtag recommendations can help users reach consensus on the adoption of hashtags, which not only controls hashtag explosion but also facilitates topic detection and tracking, hashtag-based retrieval tasks, and other hashtag related tasks.
- Different users may have different preferences for categorizing tweets. Without personalized hashtag recommendation, users will spend a lot of time on categorizing tweets and maintaining their existing classification systems.
- According to our dataset, only 20% tweets are annotated with hashtags. This means 80% of the tweets are not associated with explicit topics and they cannot be retrieved according to hashtags.
- Classical text mining methods such as Term-Frequency - Inverse Document Frequency (TF-IDF) do not get accurate results in micro blogging social networks [2]. An efficient system is needed to social network providers to recommend tags to users. To address the above problem, we have presented a Hashtag recommendation that can help to reduce the number of un-annotated tweets and also to recommend a hashtag for a given tweet using map reducing and rule mining techniques.

LITERATURE SURVEY

Recent work on hashtag recommendation for micro posts has mainly focused on two research directions: hashtag recommendation for a particular type of tweets and general purpose hashtag recommendation. In this section, we will introduce some related work.

Content-based Hashtag Recommendation

Khabiri [4] has recently proposed a content-based hashtag recommendation method, in this method, the content of a tweet, where a tweet is represented by a bag of words. The relevance between a word and a hashtag is measured on a hashtag word co-occurrence graph. The final relevance score between a tweet and a hashtag is computed as an aggregation of all the hashtag word relevance scores. However, this method cannot provide personalized results since user information is ignored.

User-level Hashtag Recommendation

Yang [5] has proposed a user-level hashtag recommendation method recently, which predicts whether or not a hashtag may be adopted by the target user in the future. Two types of features are studied: (1) Role unspecific features which describes basic characteristics of users and hashtags (e.g., the number of unique hashtags used by user u, and the number of tweets containing hashtag); and (2) role-specific features which describe the relevance between the target user u and a candidate hash tag h. However, since tweet specific information is ignored, it recommends the same set of hashtags regardless of which tweet is being considered.

Users' Dynamic Interests

In this method, a novel model, namely online Twitter-User LDA has been used to learn Twitter users' dynamic interests. Then considering the shortness, scarcity, and high volume of tweets, authors [10] introduced an effective method to discover the latent topics of streaming tweet content, which uses recently proposed incremental biterm topic model (IBTM) in which it presents an automatic hashtag recommendation method called User-IBTM by com-

bining the online Twitter-User LDA and IBTM. the experimental results of this method on real world data from Twitter showed that method based on dynamic user interests and streaming tweet content significantly outperforms several other baseline methods and can suggest more precise hashtags.

Personalized Tag Recommendation

In social tagging systems like Delicious1 and Flickr2 [12], users can annotate items with their own tags, in which case items are organized in their own way. When a user wants to annotate an item, personalized tag recommendation suggests hashtags by considering both the users' annotation preference and tags' relevance to the current item. The state of the art methods is either based on graph models or tensor factorization where annotation behavior is represented by <user, item, tag> triples. It seems that these methods can be adapted to solve our problem if we treat tweets as general items in social tagging systems. However, this cannot work for the following reason. In personalized tag recommendation, item IDs are used in graph construction or tensor factorization, which requires that items should exist in both the training set and the test set. But what we do is to recommend hashtags for new tweets instead of existing tweets. In this paper, we propose a new method to automatically recommend personalized trending hashtags based on users' tweets.

Our approach does not limit the number of candidate hashtags to be examined, but rather recommend hashtag for given tweet. Specifically, we make the following hashtag for contributions: We build an effective hashtag recommendation system using a proposed hashtag ranking method, Hashtag Frequency-Inverse Hashtag Ubiquity (HF-IHU) [8]. We provide scalable Map-Reduce algorithms to construct two fundamental structures, the term-frequency map for hashtags (THFM) and hashtag-frequency map (HFM) and then output is fed back to apriori algorithm to find the frequent words used in the given tweet, later we use rule mining techniques to recommend a best hashtag for a given tweet.

PROPOSED APPROACH

As mentioned already, our approach uses Map reducing and rule mining and ranking methods to find a particular hashtag. Map reducing function takes the tweet as input and process the given tweet to give key value pair as output. Ouput is fed back to rule mining techniques to generate a frequent item set and that is recommend as hashtag. In detail description of map reduce function and rule mining techniques are described in next section.

SYSTEM DESIGN

System Architecture

The Fig.2 represents the overall architecture of automatic hashtag recommendation system and different phases involved in recommending a hashtag for user tweet. The phases are: (i) User tweets (ii) Tweets with hashtag (iii) Map reduce programming model (iv) Frequency generation algorithm and (v) Ranking algorithm.

The phases include tweets attached with hashtag or tweets without hashtag is given as input to map reduce programming model and the map reduce model contains map, shuffle and reduce phase where as in map phase it divides an entire problem into sub problem and sends output to shuffle phase. Shuffle phase is done automatically and the sorted data is sent back to reduce phase. The output of reduce phase is key value pair is sent to frequency generation algorithm which generates certain rules and that is given as input to ranking algorithm where it recommends the correct hashtag for a given tweet.

Data Flow Diagram

The Fig.3 represents the data flow diagram of automatic hashtag recommendation system which tells how a system recommends a hash tag for a given user tweet. The data set used here the on that contains 61,732,969 tweets from 147,909 Twitter users. Data flow diagram also tells what are the phases involved in the model and inputs to each phase to process a task and also output of each phase which is given to other phase and it also informs what kind of data is sent to each phase. For example, the map phase takes input as set of tweets with hashtag and outputs the key value pair whereas shuffle phase takes input as key value pair and generates sorted data and same procedure follows for all the phases in flow diagram.

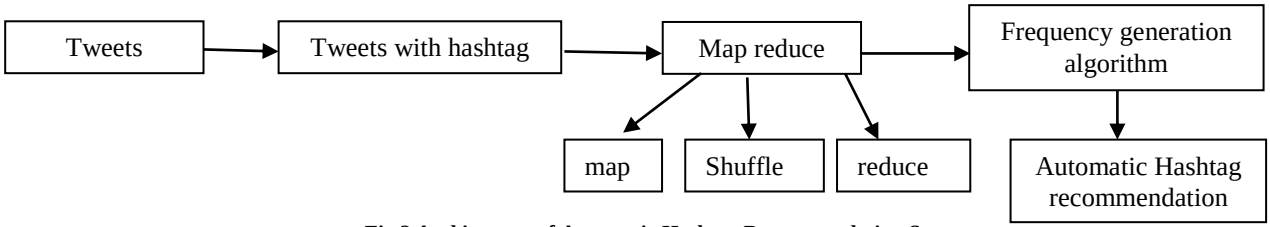

Fig.2 Architecture of Automatic Hashtag Recommendation System

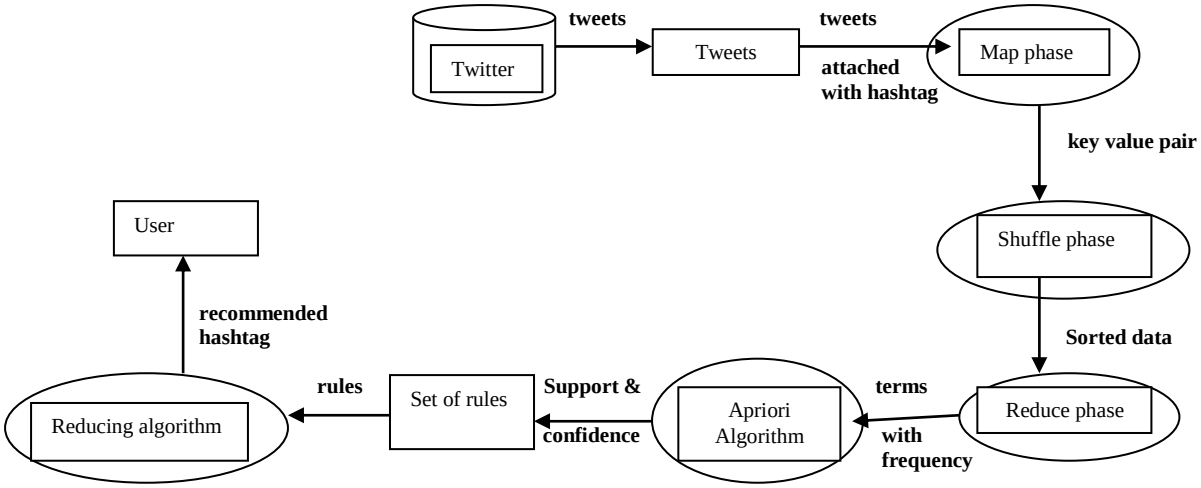

Fig.3 Data Flow Diagram

TECHNIQUES

Map Reduce Model for Hashtag Recommendation

The Map-Reduce model consists of three phases: Map, Shuffle and Reduce. Map and Reduce are user-provided functions, but Shuffle is performed automatically by the Map- Reduce framework between the Map phase and the Reduce phase**.** In map phase of map reduce model, given a data set, the master node divides the problem into data-parallel sub-problems and distributes them to worker nodes. At each worker node, the Map function produces a set of key-value pairs as intermediate outputs. Shuffle collects the intermediate outputs from the Map function and groups them by key. Worker nodes are then assigned with the grouped outputs and perform the Reduce task to process the final results. Provided the input '*government of the people by the people for the people*', Fig.4 illustrates a simple example of Map-Reduce that computes the term-frequency of the input. In the Map phase, instead of performing the Map function on the entire input, the input is divided into sub-problems and distributed to worker nodes. The Map function for this example outputs key-value pairs using terms as keys and '1' as values.

In shuffle phase the outputs from Map that is key value pair are fed into the Shuffle phase as shown in Fig 5 Shuffle sorts the data by key (terms in this example) and sends the sorted data to the Reduce phase.

In the Reduce phase, it takes the input from shuffle phase that is sorted data. The output of shuffle phase is given to reduce phase. In the reduce phase, for the given input each reduce task runs the user-provided function. The received input results are combined by key and the final output includes one value per key. In this example, from the Fig 6 the Reduce function sums the value, '1', for each key to generate a term frequency for the sample input. Fig. 4 depicts the output from the Reduce phase for the example.

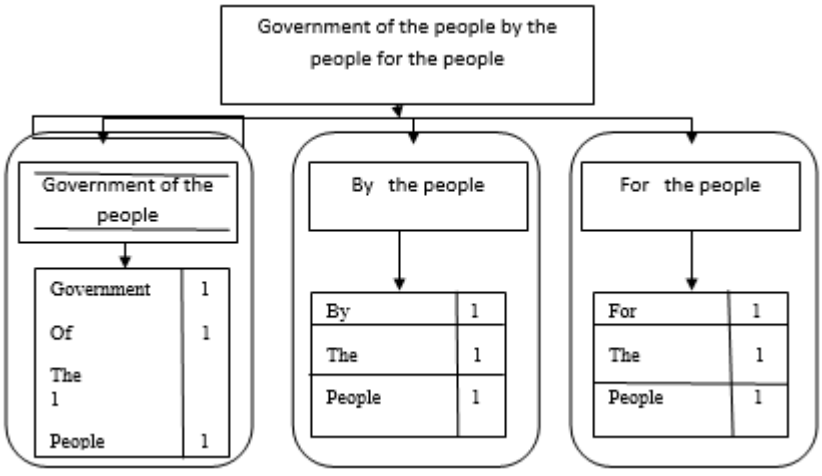

Fig.4 map reduce-map phase

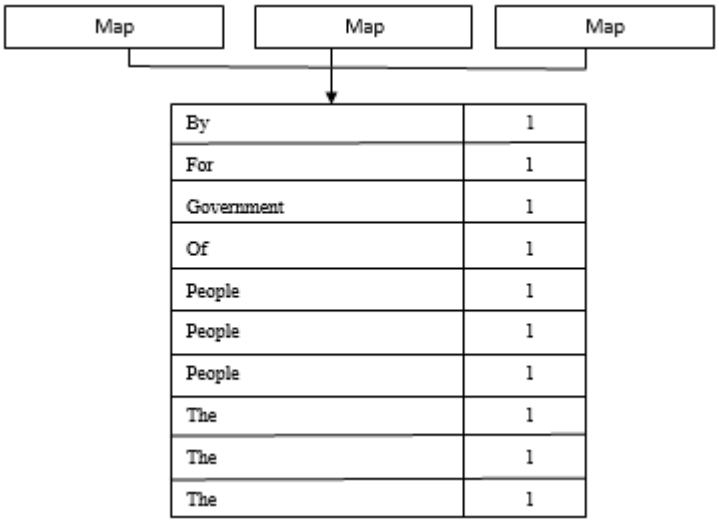

Fig.5 shuffle phase

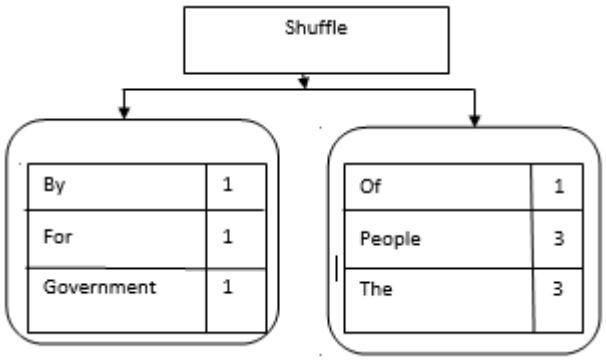

Fig.6 Reduce Phase

The Map functions are also used to generate the THFM and HFM are similar to the Map function in the above Map-Reduce example. Provided a training data set containing a list of tweets, the program prints lines in the format of #hashtag:term, where #hashtag is a hashtag appeared in a tweet and term is a term appeared in a tweet with the hashtag.

Apriori Algorithm

As soon as we get output from the map reduce function we will be able to recommend an hashtag by using the out-put from map reduce function i,e key value pair by taking the key with more value as a hashtag. But in few cases their maybe two or more keys with same value, in that case recommend all those keys as hashtag does not make sense. To overcome this conflict, we go with apriori algorithm which takes input from map function and finds the frequent item set as hashtag. The processing steps of apriori is shown below.

The Apriori Algorithm: Finding frequent Itemsets using candidate generation. Uses the prior knowledge of frequent item set.

- Apriori property:'All non-empty subsets of a frequent itemset must also be frequent.'
- k-itemsets are used to explore (k+1) item set
- L_1 – one-itemset, L_2 – 2-itemset, L_k – k-itemset , etc
- Step-1: Join step: To find L_k, a set of candidate k-itemset is generated by joining L_{k-1} with itself, denote as C_k.
- Step-2: Prune step: C_k is superset of L_k, i.e. its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k. Do subset testing.

The pseudo code for apriori is shown as-

- **Pseudo-code**

```
// Ck: Candidate itemset of size k
// Lk : frequent itemset of size k
L1 = {frequent items};
for (k = 1; Lk !=∅; k++) do
    Ck+1 = candidates generated from Lk;
   for each transaction t in database do
    increment the count of all candidates in Ck+1
  that are contained in t
endfor
   Lk+1  = candidates in Ck+1 with min_support
endfor
return ∪k Lk;
```

- For every nonempty subset s of *I*, output the rule:
  - $s \rightarrow (l - s)$,

if support_count(l) / support_count(s) >= min_conf_thres

Example:

Consider a frequent itemset: *l* = {I1, I2, I5}. What are the association rules for *l* ?

Non-empty subsets of I = {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, {I5}

Ranking Hashtags with HF-IHU

After generating frequent itemset from apriori, the next step is to score hashtags in the data set i,e itemset with highest confidence as hashtag and that hashtag is recommended for a user . But in this case also there may be conflict that two itemsets with same confidence, so to avoid these problem we go with ranking algorithm. Our proposed scoring method utilizes the variation of the TF-IDF scheme, we call Hashtag Frequency-Inverse Hashtag Ubiquity (HF-IHU). HF-IHU has two opposing weighting factors: the first is the frequency with which a hashtag appears with a given term (the hashtag frequency). The second is the hashtag ubiquity which discounts hashtags that are prevalent in all contexts and rewards hashtags that are tightly associated with a narrow subset of terms. As we go through this approaches, we will be able to recommend a better hashtag for a given tweet than other methods.

EXPERIMENTAL RESULTS

The basic statistics of the datasets will be tweets containing all the words and hash tags in tweets are stemmed and transformed to lowercase. Stop words are removed. Retweets and replies are removed, since we only focus on annotation on originally composed tweets. Due to the limited space of tweets, most URLs are shortened using short URL services. Since URLs change frequently from day\ to day [8], we only used the truncated address at the last level.

For example, 'www.bbc.co.uk/food/ page Name' is truncated to 'www.bbc.co.uk/food'. we only recommend hash tags that are learned from the training data, new hash tags are removed from the test data. Among all the tweets in the test set, only 12% tweets are annotated by brand new hash tags.

The data set used here is the one that contains 61,732,969 tweets from 147,909 Twitter users. 49,423,058 of tweets (~80%) do not have any tags and 12,309,911 (~20%) have at least one tag. The tag distribution of tweets is displayed in Fig.7[2] in which it follows a power law. It means that tags with low repetition are more than tags with high repetition.

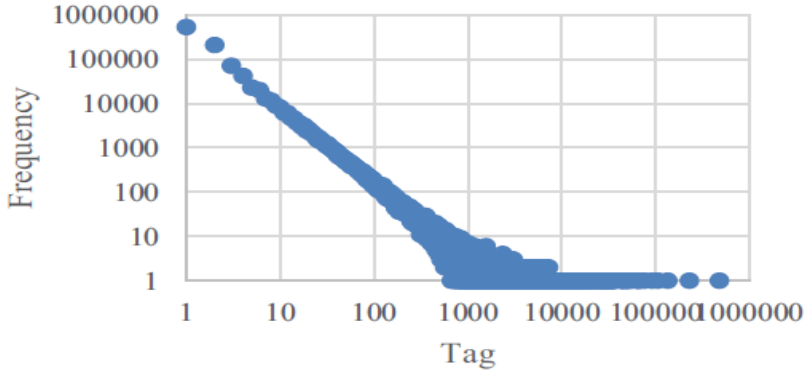

Fig.7 Tag Distribution in Data Set

CONCLUSION

The objective of this paper is to implement an effective hashtag recommendation system that automatically suggests a personalized hashtag for Twitter users. Inspired by classic information retrieval approaches, we proposed the use of a data structure to store two frequency maps that are to be built prior to performing the hashtag ranking. In this paper, we proposed a map reducing and ranking approaches for Twitter hashtag recommendation, the given tweet is processed through map, shuffle, reduce phase to generate a key value pair, then the output is again processed through rule mining techniques and ranking algorithm to find a particular hashtag for a given tweet.

Finally, we note that hashtag recommendation may be relevant for more use-cases than our work has so far explored. While our research has demonstrated promising results on recommending hash tags, the scope of the research can be extended in several other directions in the future. We discuss the most prominent. There exist several studies on sentiment analysis for the domain of micro blogs. Text sentiment could potentially be used to detect. user's interests more accurately and make better hash tag recommendations. In particular, a recommendation system used to recommend hash tags for a particular tweet from within a user's own lexicon. we conclude that our approach for recommending a particular hashtag for a given tweet is more reliable than other methods.

REFERENCES

[1] Jia Li and Hua Xu, Suggest What to Tag, Recommending More Precise Hashtags Based on Users Dynamic Interests and Streaming Tweet Content, *Knowledge Based Systems*, **2016**, 106, 196–205.
[2] Scott A Wallace, A Hashtag Recommendation System for Twitter Data Streams, *Computational Social Network, Springer*, **2016**, 3(3), 2197-2223.
[3] Frederic Godin, Towards Twitter Hash Tag Recommendation using Distributed Word Representations and a Deep Feed Forward Neural Network, *Proceeding of 24th IEEE International Conference on Advances in Computing, Communication and Informatics*, New Delhi, India, **2014**, 362-368.
[4] E Khabiri, J Caver Lee and KY Kamath, Predicting Semantic Annotations on the Real-Time Web, *Proceedings of 23rd ACM Conference on Hypertext and Social Media*, Wisconsin, USA, **2012**, 219–228.
[5] L Yang, T Sun, M Zhang and Q Mei, We Know What @you #tag: Does the Dual Role Affect Hashtag Adoption?, *WWW International World Wide Web Conference Committee* (IW3C2), **2014**, 3(3), 261–270.
[6] Wei Fang, We Can Learn Your #Hashtags: Connecting Tweets to Explicit Topics, *Proceeding of IEEE International Conference on Data Engineering*, Moscow, **2014**, 856-867.
[7] Mir Saman Tajbakhsh and Jamshid Bagherzadeh, Micro Blogging Hash Tag Recommendation System Based on Semantic TF-IDF, *Proceedings of 4th International Conference on Future Internet of Things and Cloud Workshops*, Vienna, Austria, **2016**.
[8] M Baroni, G Dinu and G Kruszewski. Dont Count, Predict! a Systematic Comparison of Context-Counting vs. Context-Predicting Semantic Vectors, *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*, Maryland, **2014**, 238-247.
[9] O Tsur and A Rappoport, Whats in a Hashtag?: Content Based Prediction of the Spread of Ideas in Micro Blogging Communities, *Proceedings of WSDM 5th International Conference on Web Search and Data Mining*, Washington, USA, **2012**, 643–652.
[10] F Godin, V Slavkovikj, W De Neve, B Schrauwen and R Van de Walle, Using topic models for Twitter Hash Tag Recommendation, *In World Wide Web 2013, Companion ACM*, **2013**, 593-596.
[11] Su Mon Kywe, tuan-annh-hoang, Fied-Zhu, On Recommending Hashtags in Twitter Networks, *Proceedings of 4th International Conference on Social Informatics*, **2012**, 337–350.
[12] W Feng and J Wang, Incorporating Heterogeneous Information for Personalized Tag Recommendation in Social Tagging Systems, KDD 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, **2012**, 1276–1284.
[13] https://twitter.com