Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 364 – 371

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Keyword Based Tweet Extraction and Detection of Related Topics

Amrutha Benny[a,]*, Mintu Philip[b]

Department of Computer Science and Engineering, Rajagiri School of Engineering & Technology, Kochi- 682 039, Kerala, India

Abstract

Twitter is a micro blogging site that helps the transfer of information as short length tweets. The large quantum of information makes it necessary to find out methods and tools to summarize them. Our research work is to propose a method, which collect tweets using a specific keyword and then, summarizes them to find out topics related to that keyword. The topic detection is done by using clusters of frequent patterns. Already existing pattern oriented topic detection techniques suffer from the wrong correlation problem of patterns. In this paper, we propose two algorithms, TDA (Topic Detection using AGF) and TCTR (Topic Clustering and Tweet Retrieval), which will help to overcome this problem. From various experimental results, it is observed that the proposed method can maintain good performance irrespective of the size of the data set.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Twitter; Topic extraction; frequent patterns; pattern association; clustering;

1. Introduction

According to the study of social networking sites in 2013, 73% of online adults are now using at least one of the social networking sites and out of this 42% use multiple networking sites[10]. From these studies, the importance of social networking sites in our everyday life is very clear. Social networking sites were initially used by people just to communicate with friends through messages. But now the role of social networking sites in the news spreading area is a non-negligible one. An important feature of social networking sites is that they provide a platform that connects

* Corresponding author. Tel.: +91- 9400457517.
E-mail address: amruthabenny@gmail.com

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.032

people from different parts of the world, thus helping to spread the information quickly. Some of the important social networking sites are Facebook, Twitter, Instagram, LinkedIn, etc.

Our research work is about the extracting information from the Twitter. Twitter is an important social networking site and a micro blog, which transfer information as short length messages known as tweets. The maximum length of one tweet is 140 characters. These short length messages increase the hardness in extraction of topics. Through sites like twitter, even normal people (non-journalists) can spread the news. One of the main Objectives of the proposed system is to summarize and cluster the tweets based upon certain keywords. The keywords can be any noun in English about which the user wants to know.

There are many methods to extract topics from Twitter. They are explained in section 2. Mainly these methods can be classified into two, tweet based clustering and pattern based clustering[11]. Both methods have their own advantages and disadvantages. In tweet based clustering method, first collect tweets using certain queries, then cluster them together and extracts topics from them. But tweet based clustering method may lead to cluster fragmentation problems. In other words, pattern based clustering method, first find out patterns, then cluster them together and find out topics from the pattern clusters. Wrong correlation of keywords is an important disadvantage of pattern based clustering techniques. So the second objective of the proposed system is to introduce a new pattern based topic detection technique, which is free from wrong correlation of patterns.

Along with satisfying the above two objectives, accuracy of results is also an important factor. In many existing topic extraction methods, accuracy of the results varying, depending on the size of data sets. So, obtaining better result accuracies, and maintaining them, irrespective of the size of input corpus is the third objective of the proposed system.

The related works that we have done is summarized in section 2. Then a detailed view of the proposed system is given in section 3. Section 4 includes various graphs to show the experimental results. Section 5 is the concluding part and we have also included our future work along with it.

2. Related work

Many research works have analyzed in the area of topic detection from tweets. As explained in the introduction section these methods can mainly be divided into two, tweet based and pattern based clustering. One example of the tweet oriented topic detection method is introduced by Swit Phuvipadawat et al.[2] According to this method first the tweets are extracted using queries like #tag which are then clustered based on the similarity. Similarity checking is done by using the term frequency (tf) value of each word in those tweets.

Vineeth Rakesh et al.[3] proposed a paper to extract location specific information from tweets. Initially, this method collects location specific tweets and then clusters them by using LDA model[6]. The disadvantage of this method is the training process used to improve the result of clustering.

The method introduced by Ahmed Rafea et al.[4], first vectored the tweets using the tf-idf (term frequency inverse document frequency – explained in section 3.2) values of words, then cluster them using k-means clustering. Because this method uses k-means clustering, the number of topics (clusters) needed to be defined earlier.

Next two methods extract topics using clusters of patterns. Hwi-Gang Kim et al.[5] used the concept that the words with the same frequencies during a period of time can be grouped together. This method needs the observation of a long period and also finds general topics like Easter, sports, etc.

The method described by Andre Klahold et al.[1] is for topic extraction from text documents. This method finds frequent patterns by using *tf-idf*, and then clustering is done by using maximum values of Associative Gravity force (AGF). The proposed system is one enhancement to this method.

All the methods explained above have their own disadvantages as we stated. The proposed system explained in section 3 can overcome all these disadvantages.

3. Proposed topic extraction method

In this section, we present a detailed description of the proposed system, which is one pattern based topic extraction method. Fig. 1 shows the flow diagram of the proposed system.

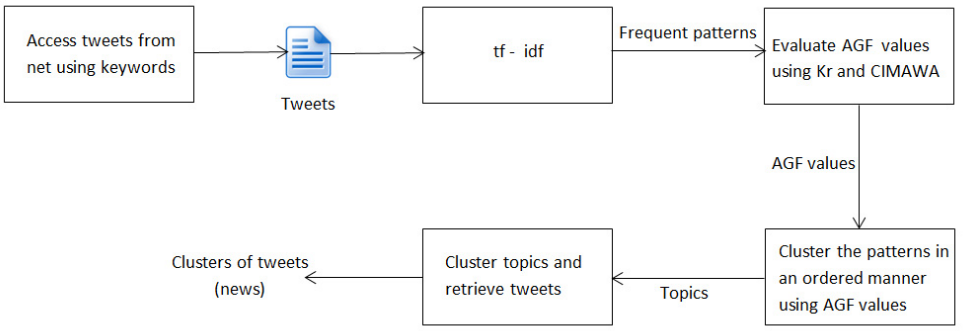

Fig. 1. Flow diagram of the proposed method.

3.1. Access tweets from net using keywords

Collecting tweets is the first process. For this, tweets are extracted online using keywords. For example, in our experiments we used "*India*" as the keyword and then collect all tweets related to that keyword. So the corpus will be a collection of tweets. For collecting tweets from net used the Twitter API.

3.2. tf – idf (Term Frequency and Inverse Document Frequency)

The proposed system is a pattern based method, so next step is the identification of frequent patterns. Frequent patterns are detected by using *tf-idf* [9]. Unlike other methods, *tf-idf* evaluates the importance of a word in one document, by considering the inverse document frequency of a word along with its frequency.

$$\text{tf-idf}(w, t, D) = \text{tf}(w, t) * \text{idf}(t, D) \quad (1)$$

$$\text{idf}(t, D) = \log \frac{T}{|\{t \in D: w \in t\}|} \quad (2)$$

Equation of *tf –idf* for word *w* is defined in (1). In (1) *t* represents one tweet and *D* stands for a document, which would be the corpus of tweets in our work. *tf(w, t)* is the term frequency, the number of occurrences, of *w* in a tweet *t*. The equation for the inverse document frequency, *idf(t, D)* is defined in (2), in which T represents the total number of tweets in document *D*.

3.3. Evaluate AGF values using Kr and CIMAWA

After finding out the frequent patterns using *tf-idf*, evaluates the AGF[1] (Associative Gravity Force) values between each pattern pairs using two other parameters, *Kr*[1] (Keyword rating) and CIMAWA[1] (Concept for the Imitation of the Mental Ability of Word Association).

3.3.1. Evaluate Keyword Rating (Kr)

$$\text{Kr}(w) = \text{f}(w) * \log \frac{T}{|\{t \in D: w \in t\}|} \quad (3)$$

Kr of each frequent word, which is obtained from *tf-idf* should evaluate using the equation defined in (3)[1]. In (3), *f(w)* represents the number of occurrences of *w*, in one document. Even though equations (1), (2) and (3) look similar, they differ in the way in which word frequency is evaluated. Word with the highest value of *Kr* will be taken as the most important one.

3.3.2. Evaluate CIMAWA

$$\mathrm{CIMAWA(x(y))}=\frac{\mathrm{Cooc(x,y)}}{\mathrm{f(y)}}+\zeta*\frac{\mathrm{Cooc(x,y)}}{\mathrm{f(x)}} \tag{4}$$

Equation for evaluating CIMAWA is given in (4)[1]. Instead of evaluating CIMAWA value for each pair of frequent patterns, *x* and *y*, CIMAWA(x(y)) is evaluated only if *y* occurs after *x* in any of the tweet.

Co-occurrence (Cooc(x, y)) is the number of tweets in which both x and y occurred together. *f(y)* represents the number of tweets in which *y* occurred, in other words, *x* in the case of *f(x)*. ζ is a damping factor, whose value is defined in the interval 0 and 1. Through various case studies it has been proved that the best value for ζ is 0.5.

3.3.3. Evaluate AGF

Kr(x) and *Kr(y)* evaluate the importance of *x* and *y* respectively in one document, and CIMAWA(x(y)) evaluates the probability of co-occurrence of *x* and *y* together.

$$\mathrm{AGF(x(y))}=\frac{\mathrm{CIMAWA(x(y))*Kr(x)}}{\mathrm{Kr(y)}} \tag{5}$$

Then next step is the evaluation of Associative Gravity Force between *x* and *y* by using the equation defined in (5)[1]. AGF evaluates the attraction between *x* and *y*. If the AGF(x(y)) is large, that means, the attraction between *x* and *y* is very high, i.e., the chances of occurrences of *x* and *y* together is very high. So we cluster the words (frequent patterns) using these AGF values.

3.4. Cluster Patterns using AGF

In this section we propose an algorithm TDA (Topic Detection using AGF), for clustering the patterns, which is shown on the Fig. 2. The aim of this algorithm is for generating the topics by clustering the frequent patterns.

TDA has three inputs M_{N*N}, *P* and AGF values. M_{N*N}, represent an *N-by-N* matrix, where *N* is the number of frequent patterns and *P* is the set of those patterns.

$$k_{ij} \in M_{N*N} = 1, \text{where } \mathrm{AGF(i(j))} = \mathrm{MAX(AGF(i(j))}\ \forall \mathrm{j}. \\ = 0, \text{ otherwise.} \tag{6}$$

Values of the matrix M_{N*N} will be generated according to the equation (6). To assign values for the matrix, M_{N*N} find out the maximum value of AGF(i(j)) ∀i, and then assign 1 to the corresponding columns. All other values will be 0. In the initial step of TDA, an empty set TOPIC is created to store all the generated topics. In step 2, find out the j^{th} pattern which have the maximum AGF(i(j)). If the set TOPIC is empty, then combine the two patterns together and add them as the first topic into TOPIC in step 3. Otherwise, if the set TOPIC is not empty, then check each topic and find out the topic in with the either p_i or p_j occurred. If such a topic obtained, then add the corresponding p_j or p_i into the L^{th} location of that topic by using the AGF values as shown in the steps from 5 to 8.

By using TDA, we created clusters of frequent patterns, which are named as topics. But these clusters may also suffer from the wrong correlation problem. The method used to avoid this problem is described in the next section.

TDA (M_{N*N}, P, AGF)

1) Initialize an empty set TOPIC.
2) for each $k_{ij} \in M_{N*N} = 1$, take corresponding patterns p_i and p_j.
3) if TOPIC is empty, then add p_i p_j as the first topic into TOPIC.
4) else for each t ∈ TOPIC
5) if $p_i \in$ t and $p_j \notin$ t
6) Add p_j into t in a location 'L' such as AGF($p_{left}(p_j)$) > AGF($p_j(p_{left})$) && AGF($p_j(p_{right})$) > AGF($p_{right}(p_j)$), where left < L and right > L.
7) else if $p_j \in$ t and $p_i \notin$ t
8) Add p_i into t in a location 'L' such as AGF($p_{left}(p_i)$) > AGF($p_i(p_{left})$) && AGF($p_i(p_{right})$) > AGF($p_{right}(p_i)$), where left < L and right > L.
9) else if $p_i \in$ t and $p_j \in$ t, then add p_i p_j as a new topic into TOPIC.

Fig. 2. Algorithm TDA.

3.5. Cluster topics and Retrieve tweets

In this section we propose another algorithm named TCTR (Topic Clustering and Tweet Retrieval), which is shown on the Fig. 3. The only one input to the algorithm is the set TOPIC, which is obtained as the output of TDR.

TCTR (TOPIC)

1) Sort the topics $t_i \in$ TOPIC, in the decreasing order of their number of words.
2) Find out the number of common words in each pair of topics.
   com = no.common_words(t_i, t_j).
3) for each $t_i \in$ TOPIC
4) for each $t_j \in$ TOPIC, where j > i
5) if com > (no.words(t_j) + 1) / 2, then cluster them together.
6) $\forall t_i \in$ TOPIC returns the tweets contains all words in t_i.
7) To reduce redundancy remove repeating tweets.

Fig. 3. Algorithm TCTR

The aim of TCTR is to cluster the topics and return the corresponding tweets as final outputs. In the first step of TCTR, sort all topics in the descending order of their number of words. The sorting is done to obtain a better clustering. Then clustering is done by finding the common number of words in each pair of topics as in the steps from 2 to 5. Step 6 will help to avoid the wrong correlation of patterns by retrieving the tweets, which contain all the words in one topic. In step 7, the repeating tweets are removed.

4. Experimental analysis

For experimental analysis, we first collect tweets that contain the keyword "India". The output of our experiments was found to be the clusters of relevant tweets related to that keyword. Andre Klahold et al.[1], tested their method across different clustering techniques and shown that the AGF clustering method is the best one. Then

to test the performance of our proposed system, we compared it with the method proposed by Andre Klahold et al.[1]. As the exact topics that should be obtained were unknown, testing of the system has been done by checking the quality of clusters. The comparison has been done based on three parameters purity, cluster entropy and class entropy. Purity will measure how accurately the tweets are clustered.

$$\text{Purity} = \frac{\text{Number of currectly clustered tweets}}{\text{Total number of tweets}} \tag{7}$$

The equation for purity is defined in equation (7). Large value of purity results in better clustering. Second parameter considered is cluster entropy[7]. This parameter checks the homogeneity of clusters. The optimal value of cluster entropy is zero. If the number of topics in one cluster increases, then the cluster entropy will also increase.

$$e_j = -\sum_i \frac{c(i,j)}{\sum_i c(i,j)} * log \frac{c(i,j)}{\sum_i c(i,j)} \tag{8}$$

$$e_j^{total} = \frac{1}{\text{m}} \sum_j e_j * [\sum_i \text{c(i,j)}] \tag{9}$$

Equation for cluster entropy is shown in (8)[1] and (9)[1]. Equation (8) defines the cluster entropy of a single cluster *j*, and (9) defines the cluster entropy of all clusters. *c(i, j)* is the number of topics *i* in a cluster *j* and m is the total number of clusters. The third parameter considered for evaluation is class entropy[8]. Class entropy evaluates how cluster fragmentation rate. Optimal value of class entropy is zero. If a single topic is present in multiple clusters, it increases the value of class entropy.

$$e_i = -\sum_j \frac{c(i,j)}{\sum_j c(i,j)} * log \frac{c(i,j)}{\sum_j c(i,j)} \tag{10}$$

$$e_i^{total} = \frac{1}{\text{m}} \sum_i e_i * [\sum_j \text{c(i,j)}] \tag{11}$$

Equation for class entropy is shown in (10)[1] and (11)[1]. Equation (10) defines the class entropy of a single topic *i*, and (11) defines the class entropy for all topics. Graphical representations of results obtained by using the parameters purity, cluster entropy and class entropy are shown in Fig. 4, 5 and 6 respectively.

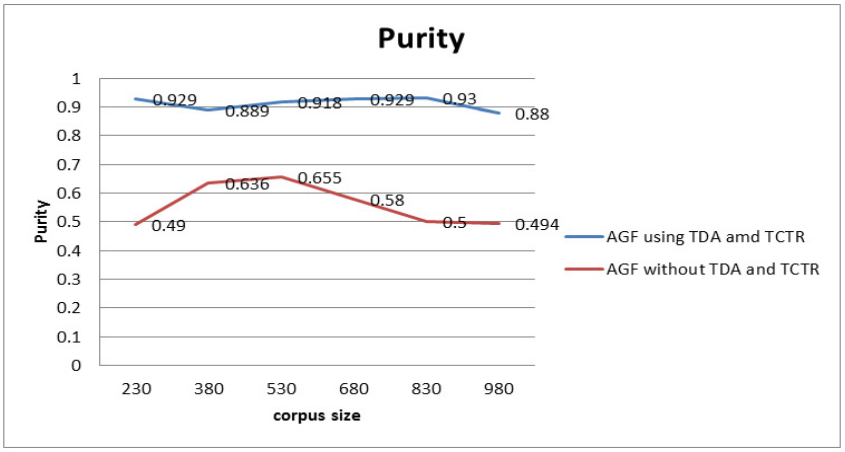

Fig. 4. Purity base comparison of proposed system and method in Andre Klahold et al.[1]

Even for different sized data sets, the proposed system can maintain the best purity in the range of 0.88 to 0.93, as depicted in Fig. 4. But purity of the method explained by Andre Klahold et al.[1] produces optimal results only for medium sized data sets. As the size of the data set falls outside the medium range, the purity decreases.

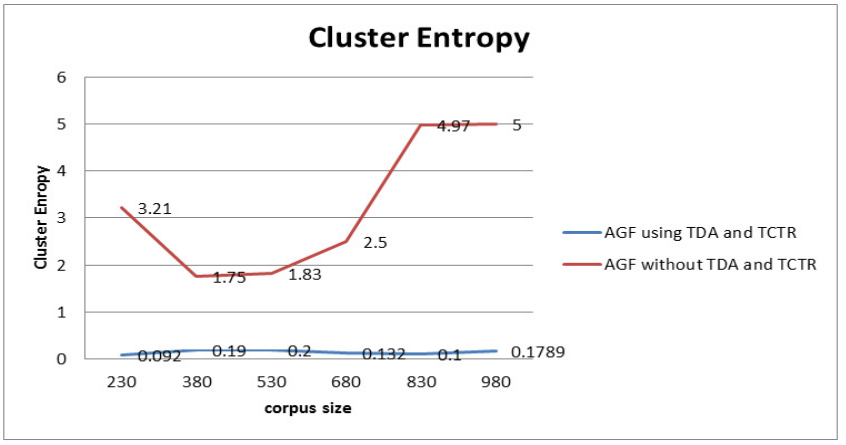

Fig. 5. Comparison of proposed system and method by Andre Klahold et al.[1] using cluster entropy.

The second parameter considered is cluster entropy[7], which is plotted in Fig. 5. The proposed system can maintain better cluster entropy for different sized data sets. Values of cluster entropy in the existing method[1] change in a manner similar to the variation of purity values.

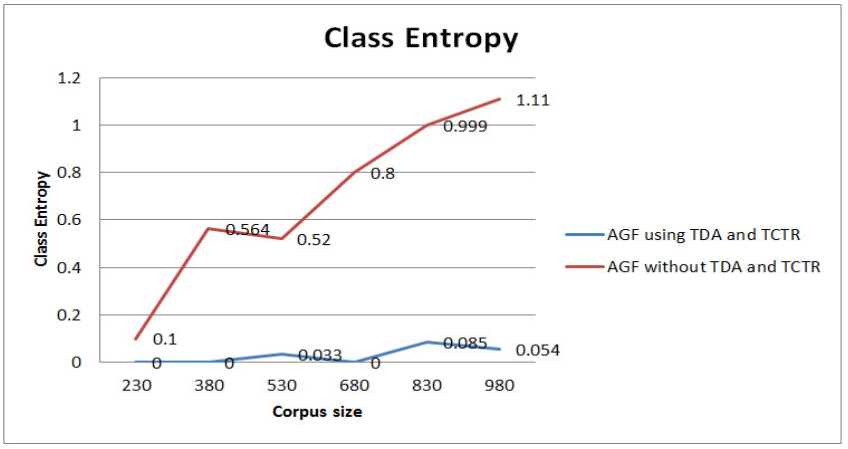

Fig. 6. Comparison of proposed system and method by Andre Klahold et al.[1] using class entropy.

Graph for class entropy of the method proposed by Andre Klahold et al[1] is nearly linear. Cluster fragmentation of the existing system is less for small sized corpus. But, as the size of corpus increases the cluster fragmentation also increases linearly. The maximum value of class entropy in the case of the proposed system is 0.085. The proposed system also shows zero cluster fragmentation for corpuses of size 230, 380 and 680.

From the above three graphs we can conclude that the proposed system has better performance with greater purity, lesser cluster and class entropy values. The proposed system can maintain good accuracy irrespective of the input corpus size, but the performance of the method proposed by Andre Klahold et al.[1] reduces when the corpus size is too large or too small.

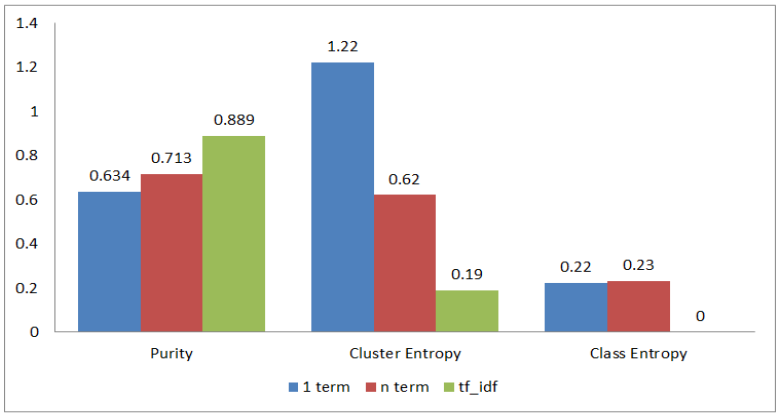

Fig. 7. Comparison of 1 term, n term and *tf-idf* approaches for pattern extraction using proposed system.

The proposed method extracts frequent patterns using *tf-idf* approach. We also analyze the performance of the *1-term* and *n-term* approaches for extracting frequent patterns in lieu of the *tf-idf* approach used in the proposed system. For comparison, we used a corpus containing 530 tweets and the result is shown in Fig. 7. From Fig. 7, it is clear that *tf-idf* outperforms the other approaches for topic detection.

4. Conclusion and future works

News spreading capability of social networking sites is really high. In order to harness this ability of social networking sites, we propose a new method to extract the topics related to certain keywords. In this paper, two novel algorithms TDA and TCTR are used to extract topics using the AGF values. The proposed system avoids the wrong correlation problem of patterns using TCTR algorithm. The various experiments conducted indicate that our system can produce better result irrespective of the corpus size.

As future works, we will consider the sentiment and emotions of tweets. The number of retweets will also be taken into consideration.

References

1. Andre Klahold, et al. *A Framework to utilize the Human Ability of Word Association for detecting Multi Topic Structures in Text Documents*. IEEE; 2013.
2. Phuvipadawat, Swit, et al. *Breaking news detection and tracking in twitter*. Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. Vol. 3. IEEE; 2010.
3. Rakesh, Vineeth, et al. *Location-specific tweet detection and topic summarization in Twitter*. In Advances in Social Networks Analysis and Mining (ASONAM). 2013 IEEE/ACM International Conference on, pp. 1441-1444. IEEE; 2013.
4. Rafea, Ahmed, et al. *Topic extraction in social media*. Collaboration Technologies and Systems (CTS), 2013 International Conference on. IEEE; 2013.
5. Hwi-Gang Kim et al., *Discovering Hot Topic using Twitter Streaming Data*. ACM International Cof. on Advances in Social Networks Analysis and Mining, IEEE; 2013.
6. R. Mateosian. *Micro Review: Twitter*. Micro, IEEE, 29, Issue 4:87–88; July-August 2009.
7. D. Boley. *Principal Direction Divisive Partitioning*. Data Mining and Knowledge discovery, 2(4), pp.325-344; 1998.
8. J. He, A.-H. Tan et al. *On Quantitative Evaluation of Clustering Systems*. Information Retrieval and Clustering, Kluwer Academic Publishers, pp.105-134; 2003.
9. A. Klahold et al. *Computation of AsymmetricalSemantic Document Relations*. Proceedings of the the 13th International Conference on Artificial Intelligence and Soft Computing, Spain; 2009.
10. PewResearch. http://www.pewinternet.org/2013/12/30/social-media-update-2013. Viewed august 2014.
11. Luca Maria Aiello et al. *Sensing Trending Topics in Twitter*. IEEE Trans. on multimedia, vol. 15, no. 6; 2013.