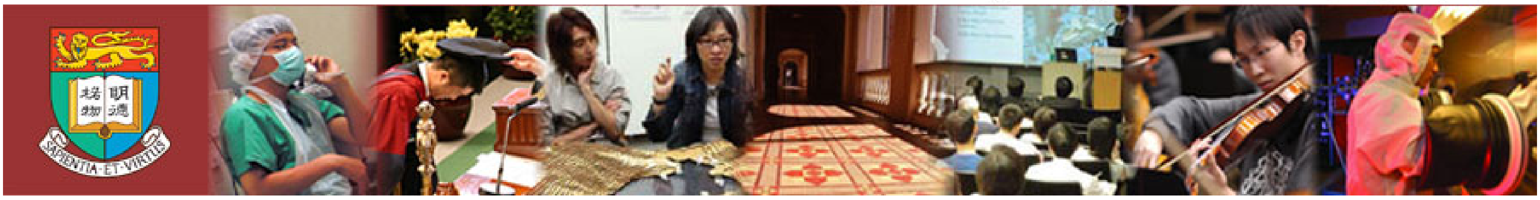

2013 IEEE 37th Annual Computer Software and Applications Conference Workshops

Towards a Better Similarity Measure for Keyword Profiling via Clustering

Y Liang[1], K.P. Chow[1], L.C.K. Hui[1], J. Fang[1,2], S.M. Yiu[1]
[1]Department of Computer Science
The University of Hong Kong, Hong Kong
[2]Department of Optoelectronic Engineering
Jinan University, China

Shuhui Hou[3]
[3]Dept. of Information and Computing Science
University of Science and Technology Beijing, China

***Abstract*— Automatic profiling for users and postings can help law enforcement units cluster and classify users and postings effectively so that potential problematic users and postings can be identified easily. A core problem in this application is to come up with effective profiles and a good measure to compare the similarity of two profiles. In this paper, we investigate an existing keyword-based user profiling scheme and identify its limitations. Then, we propose an improved version of it and demonstrate that our proposed version is more consistent than the existing approach with respect to the observed replied rates of a user to a posting based on the similarity of the profiles.**

Keywords — user profiling; keyword clustering; similarity measure

I. Introduction

With the development of Internet, forums become a popular place for the public to share knowledge and opinions, but the popularity also provides a hotbed for illegal activities. Some people use it as a platform to spread rumors, or gathering people to do things harmful to the society.

Since online forums contain a lot of valuable information, it is necessary for law enforcement to monitor it. Some countries actually assign officers to monitor the forums and check the postings manually every day. Once a problematic posting is identified, the authority will be alerted and appropriate actions may be taken. Obviously, there are some drawbacks of this method. Firstly, some posts might be deleted after a short period of time, so it is possible that when the law enforcement officer visits the page, the post is not there any more. Even the problem is reported by the public, there is no way for the officer to copy down the post (as evidence or hints for further investigation). Secondly, monitoring forums will be tedious and time consuming. Thousands of new posts are posted on the forums every day and it is impossible that the task can be done manually.

A more reasonable approach is to develop an automatic monitoring system. Such a system consists of quite a few key components. For examples: (1) crawler – automatically fetch postings every day; and (2) posting analyzer – analyze each post and identify the problematic ones. (1) is already achievable. There are many crawlers available and it is not hard to implement one once we know the format of the forum. However, for (2), the solution is still far from satisfaction. In this paper, instead of attempting to solve (2), we focus on the following related problem – how to profile a user based on his/her postings. This user profiling is very useful in computer forensics as well as Internet marketing. A user profile can capture the interests of the user successfully and allow law enforcement officers to classify whether a user should be monitored closely or allow marketing people to decide what to sell to the user. An intuitive idea to construct such a user profile automatically is to extract keywords from the postings of a user, then form a profile based on these keywords. Similarly, a post can also be characterized by the keywords used in the post (post profiling). If the profiling is done appropriately, we can classify whether a post is a target post for further investigation; and whether a user is a target user to be monitored. To classify or cluster either users or posts, we need a similarity measure on the profiles. In this paper, we try to investigate this problem. We identify some limitations of existing keyword-based user profiles. Then, we provide an improved method to construct the profile and compute the similarity of two profiles. We illustrate our results using real postings and show that our similarity measure is more consistent with the number of replies from users having a higher profile similarity as the post.

The remaining sections of the paper are organized as follows. Section II provides an overview of related existing work and the limitations of existing keyword-based user profiles. Section III introduces our proposed method. Section IV shows the experimental result and Section V concludes the paper.

II. Literature Review

A. Related Work

The concept of user profiling is not new and has been used in quite a number of areas such as personalized marketing and forming user groups in various applications [1]. For examples, Hotminer [2] has been used in HP to extract useful information in grouping users or collecting statistics by profiling how users navigate their customer service website. [3] provides another example on how to profile users based on what mouse operations they use. A method to extract and maintain user profiles from large scale data with high quality and efficiency is provided in [4]. [5] is another example on forming user

978-0-7695-4987-3/13 $26.00 © 2013 IEEE
DOI 10.1109/COMPSACW.2013.19

profiles based on the characteristics of mobile phone calls made by users in order to partition users into different categories, for say marketing purposes.

On the other hand, there are some other works that try to identify hot topics or current popular events based on publicly available information. For examples, BlogPlus [6] is a website that that shows current popular event in blogs. Detecting hot topics for online forums is demonstrated in [7]. However, there are very little work that tries to profile users based only on their postings in online forums (except only [8]).

B. Automatic Online Monitoring and Data-Mining Internet Forums

As mentioned in the above, [8] provides the first step in profiling users based on their online postings. An automatic online monitoring and data-mining engine is proposed. In their approach, the engine will extract a list of keywords based on the contents and the topics of postings from users. Based on these keywords, they provided a method to form a keyword vector to characteristic each user (called user vector) as well as each topic (called topic vector). Then, it uses cosine similarity to predict user's interest. They showed some interesting results based on real data. The idea is promising, however, there are some technical shortcomings in their proposed system. We list them in the followings:

- Some of the keywords in the postings are double counted. This affects the accuracy of the keyword vector.
- User profiles are accumulative. Users' interests may change. By accumulating old postings in forming the user topic may not reflect the current interest of the user clearly. It will affect the prediction of users' interest.
- The similarity measurement is not very precise and is not consistent with the observed replies from the users on the postings.

C. Our Contributions

In this paper, in order to get a more precise measurement of similarity between forum user and forum topic, we suggest the followings:

- We resolve the keyword double-counting problem.
- Existing approaches rely on exact keywords to relate a profile to the other. However, in practice, even if the keywords are not exactly the same, they may be closely related. Thus, we employ a clustering technique to group keywords into clusters so that keywords in the same clusters can be considered more related than keywords appearing in different clusters.
- We introduce a time window when constructing user profiles, thus keeping the profile to capture the current interest of the user.
- We define a new measurement of similarity between forum user and forum topic.

III. Model and Methodology

The collected data including posts, topic and user information in a specific time range, will be analyzed day by day. The similarity between user and topic will be calculated with statistics of existing data, which predicts how much the user is interested in the topic. Note that we use forums that contain Chinese characters as our targets, so in the followings, the examples will contain Chinese characters, but the exact meaning of the characters is not important. Readers who cannot understand Chinese can just ignore the meaning of the words.

A. Extracting Keyword

The first step is extracting keywords from the collected data. The way of getting keyword list is similar to that in [8]. Let W_n denote the weight of an n-adjacent-character (n-adjacent-character is a word with n tokens, n from 2 to 6) and k denote a threshold (k=20) respectively. For data of the N^{th} day, we consider that a keyword appears in the N^{th} day if $W_n > k$. The complete keyword list of the N^{th} day includes new keywords appearing in the N^{th} day, keywords appearing in the past and words in a dictionary which contains some important words.

Keyword prefix might cause problems in keyword extraction. For example, if both "北京" and "北京大学" satisfy the keyword selecting rule, which one will be counted as keyword? The solution provided in [8] works as follows. Let W_{n+i} ($i \geqslant 1$)denote the weight of an (n+i)-adjacent-character and T_k denote the threshold rate (T_k =0.7). It chooses the (n+i)-adjacent-character if $W_{n+i} > T_k \times W_n$. Otherwise, it chooses the n-adjacent-character. Unfortunately, this solution inevitably leads to double count problem with the keyword accumulating day by day. Take the case of "北京" and "北京大学" as an example. One day "北京" is counted as keyword, the next day "北京大学" is counted as keyword and later both of them may be counted as keywords. In other words, double count issue will occur when generating topic profile and user profile.

In this paper, we try to solve the above double count issue when calculating similarity between user and topic instead of handling it in the keyword extraction stage.

B. Clustering Keyword

In this stage, we express every keyword as a vector and based on their vector forms, we cluster the keywords by K-means.

Suppose that the complete keyword list of the N^{th} day is $L_N=\{K_1, K_2,\ldots, K_m\}$ and K_A is an arbitrary keyword. The vector of K_A is defined as $V_{KA}=(R_{A1}, R_{A2},\ldots, R_{Am})$. The R_{Aj} ($1 \leqslant j \leqslant m$) is the relevancy of K_A with K_j and calculated by

$$R_{Aj} = F_{Aj} \times W(K_j) \qquad (1)$$

where F_{Aj} is the frequency of K_A and K_j appearing in the same topic and $W(K_j)$ is the weight of K_j which can be calculated by inverse topic frequency as follows

$$W(K_j) = \log\left(\frac{|T|}{|\{t \in T|k \in t\}|}\right) \quad (2)$$

$|T|$ is the total number of topics and $|\{t \in T|k \in t\}|$ is the number of topics hit by K_j. One month of data will be used as training set when calculating $W(K_j)$.

The keyword profile of the N^{th} day consists of the above-described keyword vectors and it is also the input of k-means clustering. The first level of k-means clustering is performed by grouping the keyword vectors into N_C ($=\frac{m}{80}$) clusters. We set the number to be 80 due to the observation: there are around 80 instances in each cluster if the clustering can be evenly performed. Therefore, we consider that 80 is an appropriate size of a keyword cluster. However, the clustering results are not even under most circumstance. Sometime the cluster size is less than 80 and sometime the cluster size is far greater than 80. For the clusters whose size is greater than 50, we will perform second level of k-means clustering. That is, one cluster will be further divided into $\left(\frac{S}{50} + 1\right)$ sub-clusters as its size s >50, where 50 is also an empirical value.

C. Preparing User Vector

For calculating the similarity between topic and user, we need to prepare user profile and topic profile.

Every user will be represented by a user vector as his profile. Let $L_U=\{K_{U1}, K_{U2},\ldots, K_{Um'}\}$ denote the keyword list hit by the user U in the N^{th} day. The user vector is defined as $V_U=(W_{U1}, W_{U2},\ldots, W_{Um'})$. The W_{Uj} ($1 \leqslant j \leqslant m'$) is the weight of the keyword K_{Ui} and calculated by

$$W_{Uj} = RF_{Uj} \times W'(K_{Uj}) \quad (3)$$

where RF_{kj} is the refined frequency of K_{Uj}. We use the refined frequency instead of original frequency F_o for lessening the impact of word frequency which will become very large as the word is used very frequently. The RF_{Uj} is calculated by

$$RF_{Uj} = \sum_{i=1}^{n'} (F_o - i \times \overline{F}) \times (1 - 0.2i) \quad (4)$$

Here, $\overline{F}$ is the average word frequency of the $(N-1)^{th}$ day, $n' = \mathrm{Min}\left(\frac{F_o}{\overline{F}}, 4\right)$. $W'(K_{Uj})$ is the weight of K_{Uj} and can be calculated by inverse user frequency below.

$$W'(K_j) = \log\left(\frac{|U|}{|\{u \in U|t \in u\}|}\right) \quad (5)$$

$|U|$ is total number of users, $|\{u \in U|t \in u\}|$ is the number of users who used this keyword.

We introduce a time window when preparing user vector, that is, we just count keywords hit by the user in a specific time range (within latest 14 days).

D. Preparing Topic Vector

Similar to user profile, every topic is also represented by a topic vector as its profile. Let $L_T=\{K_{T1}, K_{T2},\ldots, K_{Tm''}\}$ denote the keyword list hit by the topic T in the N^{th} day. The topic vector is defined as $V_T=(W_{T1}, W_{T2},\ldots, W_{Tm''})$. The W_{Tj} ($1 \leqslant j \leqslant m''$) is the weight of the keyword K_{Ti} and calculated by Tf-idf as follows.

$$W_{Tj} = \frac{tf}{m''} \times \log\left(\frac{|T|}{|\{t \in T|k \in t\}|}\right) \quad (6)$$

tf is the word frequency in the topic, $|T|$ is total number of topics and $|\{t \in T|k \in t\}|$ is the number of topics hit by this keyword.

We deal with the double count issue in this stage. As both n-adjacent-character keyword and (n+i)-adjacent-character keyword are hit by a topic, only the longer one ((n+i)-adjacent-character keyword) is counted as keyword in this topic profile.

E. Calculating Similarity between Topic and User

The similarity between topic and user is used to evaluate their relevancy. Given a user vector $V_U=(W_{U1}, W_{U2},\ldots, W_{Um'})$ and a topic vector $V_T=(W_{T1}, W_{T2},\ldots, W_{Tm''})$, we first calculate their un-normalized score, then normalize the score and obtain the similarity from the normalized score.

1) Similarity without keyword clustering

When calculating the un-normalized score of similarity without keyword clustering, we merely take keywords hit by both topic and user into consideration. We project the user vector V_U to V_U' base on the topic vector V_T, that is,

$$V_U' = (W_{U1}', W_{U2}',\ldots, W_{Um''}') \quad (7)$$

$W_{Uj}' = W_{Uj}$ ($1 \leqslant j \leqslant m''$) if the keyword corresponding to W_{Tj} is also hit by the user in the time window. Otherwise, $W_{Uj}' = 0$ ($1 \leqslant j \leqslant m''$). The un-normalized score is calculated by

$$\text{Un\_S\_withoutK} = \sum_{i=1}^{m''} W'_{Ui} \times W_{Ti} \quad (8)$$

2) Similarity with one-level keyword clustering

When calculating the un-normalized score of similarity with one-level keyword clustering, we consider not only the keywords hit by both user and topic, but also other keywords hit by the user. Assume that two different keywords K_A and K_B are grouped into the same cluster by k-means, where K_A is hit by the user but not hit by the topic and K_B is hit by the topic. The keyword K_A also contributes to the un-normalized score which is calculated by

$$\text{Un\_S\_1\_level\_Cluster} = f_1 \times \sum_{i=1}^{m''} \sum_{j=1}^{n_1} W_{Ti} \times W_{Uj} \quad (9)$$

f_1 is an impact factor of one-level clustering and we set it as 0.5 in our experiment. n_1 equals to the number of keywords in V_U grouped into the same cluster with K_{Ti}.

For clusters with larger size (e.g., size>50), we calculate the un-normalized score with two-level keyword clustering rather than one-level keyword clustering due to that the relevancy between two instances is relatively weak.

3) Similarity with two-level keyword clustering

As above-mentioned, the cluster will be further divided into sub-clusters when its size is greater than 50. We calculate un-normalized score of similarity with 2 level keyword clustering below.

$$\text{Un-S\_2\_level} = \text{Un\_S\_1\_level} + \text{Un\_S\_ bigCluster\_l1}$$

$$+\text{Un\_S\_ bigCluster\_l2} \tag{10}$$

Assume that keyword K_A is hit by the user and K_B is hit by the topic, and they fall into the same cluster (its size is greater than 50). Un_S_bigCluster_l1 is to evaluate the contribution of user keyword K_A when K_A and K_B fall into the same cluster but different sub-cluster. Un_S_bigCluster_l2 is to evaluate the contribution of user keyword K_A when K_A and K_B fall into the same sub-cluster. Un_S_bigCluster_l1 is calculated by

$$\text{Un\_S\_ bigCluster\_l1} = f_1{}' \times \sum_{i\prime=1}^{m_2} \sum_{j\prime=1}^{n_2} W_{Ti\prime} \times \mathrm{W}_{Uj\prime} \tag{11}$$

and Un_S_bigCluster_l2 is calculated by

$$\text{Un\_S\_ bigCluster\_l2} = \mathrm{f}_2{}' \times \sum_{\mathrm{i"}=1}^{m_3} \sum_{\mathrm{j"}=1}^{\mathrm{n}_3} W_{Ti"} \times \mathrm{W}_{\mathrm{Uj"}} \tag{12}$$

We set the impact factor $f_1{}'$=0.15 and f2'=0.05 in our experiment.

After calculating the un-normalized score, we normalize the score by computing

$$\text{Normalized score s=}\frac{\text{Un−normalized score}}{2\times\bar{S}\times\overline{W(k)}} \tag{13}$$

and define similarity as

$$\text{Similarity} = \begin{cases} \mathrm{S}\ (s<1) \\ 1 (s\geq1) \end{cases} \tag{14}$$

$\bar{S}$ is the average un-normalized score of the (N-1)th day and $\overline{W(k)}$ is the average sum of keyword weight of a topic of the (N-1)th day.

IV. Experimental Results

A. Introduction to dataset

The dataset is from 2013-01-01 to 2013-04-30 (total # of users : 5181).

Data from 2013-01-01 to 2013-03-08 is used to train the model (# of topics: 3054; # of posts:28724).

Data from 2013-03-08 to 2013-04-30 is used to test our methodology (# of topics: 3011; # of posts:34354).

B. One-Level K-means clustering

In One-Level K-means clustering, we show the clustering results of day 2013-03-08, where 2315 instances are divided into 28 clusters and around 64% of instances fall into the same cluster. The relevancy of instances within this large cluster is not so tight as that in the smaller clusters. We randomly pick out two clustering results of day 2013-03-08 and list them in Table 1. Note that the texts are Chinese characters and their exact meanings are not important here.

Table 1: Examples in one-level keyword clustering

Cluster 1	Cluster 2
内地, 港人, 社会, 表示, 香港人	北大, 北大教, 北大教授, 大学, 大教, 大教授, 孔庆, 孔庆东, 庆东. 教授, 教授孔庆, 教授孔庆东, 言论

C. Two-Level K-means clustering

As mentioned above, some clusters may contain a lot of instances. For such clusters, we further divide them into sub-clusters. Table 2 shows a two-level clustering result which is randomly picked from the results of day 2013-03-08.

Table 2: Examples in two-level keyword clustering

Level 1 Cluster	中央, 争取, 僭建, 反对, 国际, 投票, 提名, 特首选, 特首选举, 选特, 选特首, 个提名, 会主席, 何俊仁, 俊仁, 入闸, 刘淑仪, 区议, 危机, 参选人, 参选特首, 叶刘, 叶刘淑仪, 叶太, 名参选, 唐梁, 唐营, 商界, 子选举, 宣布, 工程, 建制, 建制派, 建筑, 建联, 当年, 当选, 报名, 挺唐, 提名票, 政纲, 新民党, 昨天, 显示, 曝光, 曾钰, 曾钰成, 最新, 民党, 民建联, 民意, 民望, 泛民, 特区, 特区政府, 特权, 研究, 能力, 行政长官, 表态, 西九, 规划, 诚信, 钰成, 政纲, 新民党, 显示, 曝光
Level 2 Cluster 1	中央, 争取, 僭建, 反对, 国际, 投票, 提名, 特首选, 特首选举, 选特, 选特首
Level 2 Cluster 2	个提名, 会主席, 何俊仁, 俊仁, 入闸, 刘淑仪, 区议, 危机, 参选人, 参选特首, 叶刘, 叶刘淑仪, 叶太, 名参选, 唐梁, 唐营, 商界, 子选举, 宣布, 工程, 建制, 建制派, 建筑, 建联, 当年, 当选, 报名, 挺唐, 提名票, 政纲, 新民党, 昨天, 显示, 曝光, 曾钰, 曾钰成, 最新, 民党, 民建联, 民意, 民望, 泛民, 特区, 特区政府, 特权, 研究, 能力, 行政长官, 表态, 西九, 规划,

	诚信, 钰成, 政纲, 新民党, 显示, 曝光

D. *Comparison of Similarity*

We evaluate the result by reply proportion. Firstly, we categorize the calculated similarities to 21 points from 0.0 to 1.0 with step 0.05. Then count the number of instance of each point C_p to get a normalized nor-C_p which is the reply count when the similarity distribution of each point is even. The reply proportion P_R is calculated as follow:

$$P_R = \frac{nor-C_p}{\sum_{i=1}^{21} C_{pi}} \qquad (15)$$

Our experimental results are shown in Figure 1, where the charts demonstrate the relationship between the similarity (between user profile and posting profile) and reply proportion. The x-axis denotes the value of similarity and y-axis denotes the reply proportion. The blue curve represents the result without keyword clustering, the green curve represents the result with one-level keyword clustering and the red curve represents the result with two-level keyword clustering.

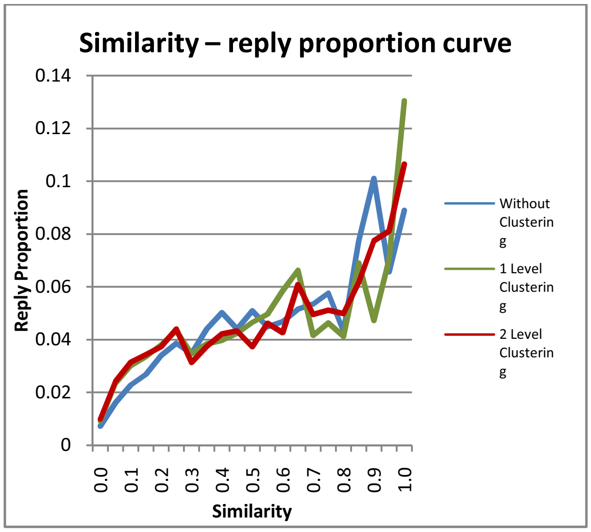

Figure1: Similarity – reply proportion curve

From the charts, we can see that only less than 1% of the replied posts with similarity 0.0 while around 10% of the replied posts with similarity 1.0. The increase of similarity is consistent with the increase of reply proportion and the trend of similarity with two-level clustering is the most stable among the three types.

V. Conclusions

In this paper, we have enhanced the method of predicting user interest in internet forums, keyword clustering. We also introduced formulae for calculating similarities. The experimental results show that the increase of reply proportion is consistent with the increase of similarity and two-level keyword clustering can make the trend more stable. But the keyword clustering result is not good enough for those keywords which have relative weak relevancy. We will improve it further in our future work.

References

[1] Susan Gauch, Micro Speretta, Aravind Chandramouli, and Alessandro Micarelli, "User Profiles for Personlaized Information Access", The adaptive web, p54-89, 2007.

[2] M. Castellanos, "Hotminer: Discovering hot topics from dirty text", Suevery of Text Mining: Clustering, Classification, and Retrieval, p123, 2004.

[3] Yoshinori Hijikata, "Implicit User Profiling for On Demand Relevance Feedback", 9th international conference on Intelligent user interfaces, P198-205, 2004.

[4] Michal Shmueli-Scheuer, Haggai Roitman, David Carmel and Yosi Mass, David Konopnicki, "Extracting User Profile from Large Scale Data". 2010 Workshop on Massive Data Analytics on the Cloud, 2010.

[5] Barbara Furletti, Lorenzo Gabrielli, Salvatore Rinzivillo, "Identifying user profile from mobile cell habits", ACM SIGKDD International Workshop on Urban Computing, p17-24, 2012.

[6] N.Glance, M. Hurst, and T. Tomokiyo, "Blogpulse: Automated trend discovery for weblogs," vol.2004, Citeseer, 2004.

[7] Nan Li, Desheng Dash Wu, "Using text mining and sentiment analysis for online forums hotspot detection and forecast", Decision Support Systems, 2010, p354-368.

[8] Y.M. Lai, Xueling Zheng, K.P Chow, Lucas C.K. Hui, S.M. Yiu, Automatic Online Monitoring and Data-Mining Internet Forums, Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2011.