Computer Science and Information Systems 11(3):1157–1172 DOI: 10.2298/CSIS130927070Z

Weibo Clustering: A New Approach Utilizing Users' Reposting Data in Social Networking Services

Guangzhi Zhang[1], Yunchuan Sun[2], Mengling Xu[1], and Rongfang Bie[1]

[1]College of Information Science and Technology, Beijing Normal University, 100875 Beijing, China
{zgz, xml}@mail.bnu.edu.cn, rfbie@bnu.edu.cn
[2]Business School, Beijing Normal University, 100875 Beijing, China
yunch@bnu.edu.cn

Abstract. As one of the most popular Social Networking Services (SNS) in China, Weibo is generating massive contents, relations and users' behavior data. Many challenges exist in how to analyze Weibo data. Most works focus on Weibo clustering and topic classification based on analyzing the text contents only. However, the traditional approaches do not work well because most messages on Weibo are very short Chinese sentences. This paper aims to propose a new approach to cluster the Weibo data by analyzing the users' reposting behavior data besides the text contents. To verify the proposed approach, a data set of users' real behaviors from the actual SNS platform is utilized. Experimental results show that the proposed method works better than previous works which depend on the text analysis only.

Keywords: behavior data, clustering, data mining, microblog, Weibo, Social Networking Services.

1. Introduction

Social Networking Services (SNS) are changing the world. In the era of Web 1.0, most netizens are just tourists to retrieve information from the Internet. Nowadays, this is not the case. Massive messages are generated by the netizens and massive public highlighting opinions are emerging. The era of "Information Explosion" has been transformed in to that of "Opinion Explosion" with the support of Social Networking Services, such as Microblog, Weibo (a kind of microblog in China), and etc. The content on the Internet, such as the text, image, audio and video, and etc., is the primary resource in the "Information" era. However, for one "opinion", only content is not enough [1, 2]. The social relation (e.g. follow, group, etc.) and users' behaviors (e.g. repost, comment, "@", etc.) play more important roles in forming an "opinion".

Most works on SNS are based on analyzing the text contents for there have been numerous of successful approaches on text mining. Unfortunately, these traditional approaches which are designed to process normative and long enough texts don't work well on SNS platforms because most of the messages are short texts. Even more, there are many kinds of data besides short texts on SNS platforms which can't be processed

with these traditional approaches [3]. Further, many hot messages on SNS even have no text, but only image or video etc.

Social links are more important in forming an opinion on the SNS platform. A schema theory is proposed to help the semantic analysis for the links among objects in [4, 5], which can be utilized in SNS platform. Social relations are more and more frequently used in recent researches and applications [6]. However, there're two problems about social relations. 1) It is hard to discover all social relations among users for its high dynamic changing and sometimes the overall relations are needed in analysis. 2) Social relations are somewhat "static"- it's somewhat "inharmonious" when compared with SNS's highly variable "dynamic". It would be more exciting if new "helper" like relations could be found [7].

In the SNS society, opinions are gradually formed in the dissemination process and every behavior of users contributes to this process. Indeed, we can construct the Web of opinions by extracting opinions from the users' behavior data [8], where opinions can be regarded as events correspondingly. Reposting is a strong opinion expression in SNS (especially in microblog); because it shows that users have a strong wish to recommend the reposted messages to their friends. In other words, one person reposting a message shows his/her strong interest on the topic.

This paper proposes a method to cluster the Weibo messages, utilizing users' interest distribution in different messages which is mined from the reposting data. The experiment results show it performs better than traditional works. The paper is structured as follows. Section 2 introduces the related works and Section 3 introduces the technology background and proposes a new method to cluster the Weibo data. Experiments and analysis are presented in Section 4. Finally, we conclude the paper in Section 5.

2. Related Works

Clustering is to organize data into sensible clusters, and is one of the most fundamental modes for understanding and learning a data set. K-means is one of the well-known and simple clustering algorithms proposed 50 years ago. In last decades, some useful research directions, such as semi-supervised clustering, ensemble clustering and so on, have been proposed [9]. K-means++ improves both the accuracy and speed of K-means by choosing the initial seeds, which satisfies users better in some specific fields [10]. In fact, K-means++ is exactly the vital inspiration of our new proposed algorithms.

TF-IDF scheme proposed by Salton and McGill in 1983 [11], is widely used to characterize documents information retrieval systems based on the vector space model. Many classical and modified TF-IDF based approaches were presented for text mining in various fields, such as topic detection and tracking in [12] (proposing a term frequency smoothing method which weaves time slices) and [13] (presenting a multi-document summarizer, which generates summaries using cluster centroids), web pages retrieval [14] (proposing several approaches to refining the TF-IDF by using one page's hyperlinked neighboring pages), image detection [15], and object matching in Google videos [16] and so on. Especially, [17] proposes a perspective of TF-IDF measures for text categorization based on term weighting theories and information theory. There are also lots of researches based on TF-IDF for different purposes, such as introducing

multi-language knowledge integration into social media datasets from Facebook and Twitter for clustering [18, 19] (enriching data representation by employing machine translation to increase the number of features from different languages, but it's useless for Chinese microblogs because of the metaphor and social background), quality-biased ranking for the high-quality contents by a regression approach which incorporates various features [20], content summarization from these collections of posts on a specific topic [21], feature selection for microblog mining [22], real-time topical news recommendation [23], hash-tag retrieval [24] (they all require the relative standard format) and so on.

LDA (Latent Dirichlet Allocation), a generative probabilistic model using TF-IDF for collections of discrete data, is a quite popular model for microblog mining [25]. Reference [26] characterizes microblogs with topic models based on "Labeled LDA", a partially supervised learning model. A modified model called "MB-LDA" is proposed on topic mining in [27], which introduces the "@" and "RT" (Retweet, Repost) into the LDA model to mine the latent relations in the conversations whose test data come from Twitter in English. Short text in microblogs brings big challenges to microblog mining utilizing traditional methods. Reference [28] proposes a method based on hidden topics analysis and text clustering to discover news topics in microblogs. Although the experimental results show this method works well on large-scale microblog dataset, the small length of news in microblog cannot ensure completeness of the whole event.

Some other literatures put forward many creative ways to cluster microblog, including using semantic knowledge [4, 5 and 29] and affinity propagation [30]. Using the results of clustering, many more interesting works have been done to deepen the research on microblog, such as identifying topical authorities [31].

As a typical measurement, TF-IDF earns big success in many fields, including microblog mining. The TF-IDF based K-means algorithms also work quite well in microblog clustering. This paper deploys a clustering framework for microblog clustering based on K-means++， and propose a new RepSim measurement to measure its distance. To test the effectiveness of the proposed method we take the TF-IDF for comparison on the same data set with the same indicators.

3. Methods and Design

Microblog data is a kind of typical big data, including contents, relations, and users' behavior records. Considering the features of big data, approaches aiming to do something with the microblog data should be high-efficiency and simple enough (remember the saying "Keep It Simple and Stupid"). In this paper we attempt to find out a measurement for clustering to represent the similarity between two microblogs, which are effective and simple.

After some previous experiments, we find that the users' reposting records data meet our expectation. We here define a new "RepSim" (Reposting Similarity) distance measurement for the similarity computation between Weibos using the users' reposting records data without considering the contents of the Weibo itself, employ K-means++ to cluster Weibo data, while carefully choosing its initial centers, and then we randomly select 100 hot microblogs posted recently from Weibo for the effectiveness test.

Meanwhile, TF-IDF is applied to the same dataset, to compare with the RepSim's results. Three indicators, Cosine, Jaccard and Tanimoto, are used to evaluate the effectiveness of proposed method.

We describe our framework in detail in this part.

3.1. Clustering Framework Based on K-means++

The K-means method is a widely used clustering technique that seeks to minimize the average squared distance between two points in the same cluster. Its simplicity and speed are very appealing in practice, but it cannot guarantee general accuracy currently. K-means++ improves both the accuracy and speed of K-means by choosing the initial seeds. We propose that our clustering framework is based on K-means++, choosing the initial seeds according to author's experience with the aim to make sure the results more stable and credible, and is also relatively fair to TF-IDF and RepSim at the same time.

The K-means++ technological process is shown as follows:

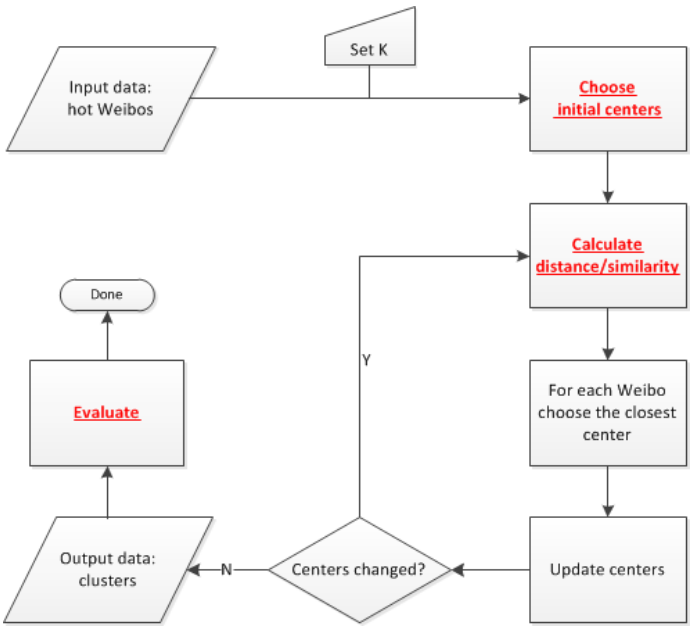

Fig. 1. The flow chart of K-means++ used in this paper

The three key points for the algorithm is how to choose the k initial centers, what the similarity or distance definition is, and how to evaluate the clustering effectiveness, which are colored red in Fig. 1.

3.2. New Similarity Measurement "RepSim"

New Proposed "RepSim" means "Reposting Similarity", which calculates the degree of similarity between microblog M_i and M_j via the ratio of shared people in all who have reposted the two microblogs. As mentioned above, "reposting" stands for "interest". Meanwhile, one person holds his/her interests stable relatively during a certain period. According to the survey about reposting, it is true that a person is interested in a microblog if he/she reposts it, and two microblogs might have something in common if both of them are reposted by one person. So it has great probability that the two microblogs belong to one cluster when clustering the set of microblogs. That means, the more reposting people M_i and M_j share, the higher probability the two microblogs have the similar topics or characteristics. Hence RepSim can measure the Weibo's similarity from the perspective of probability. We define RepSim as following:

$$\mathrm{RepSim}_{i,j} = \frac{|R_i \cap R_j|}{\sqrt[2]{|R_i| * |R_j|}}. \quad (1)$$

$R_i(R_j)$ is the set of people who repostM_i (M_j).We use square root in the denominator so as to process the balance of huge difference between their reposting times.

For example, there aretwo microblog messages reposted by people, R_1 = {A, B, C, D, E}, R_2 = {C, E, F}, we can calculate RepSim of the two messages by:

$$\mathrm{RepSim}_{1,2} = \frac{|\mathrm{R}_1 \cap \mathrm{R}_2|}{\sqrt[2]{|\mathrm{R}_1| * |\mathrm{R}_2|}} = \frac{|\{\mathrm{C}, \mathrm{E}\}|}{\sqrt[2]{5 \times 3}} = \frac{2}{\sqrt[2]{15}} \approx 0.5164$$

In fact, RepSim performs quite differently between different microblogs. The following scatter-gram shows the distribution on the 100 hot microblogs dataset.

Fig. 2 shows the distribution of RepSim of the 100 hot microblogs dataset, and there are several points to note: 1) RepSim between most microblogs (over 91%) is less than 1%; 2) we divide the 1% into 10 parts with the step of 0.1%, so the distribution through the histogram in the right side is relatively homogeneous; 3) it's hard to get a large RepSim, but RepSim has a clear discrimination for Weibo clustering.

Classical "TF-IDF" is a widely used method to characterize documents information retrieval systems based on the vector space model. TF-IDF is a notable measurement to express the similarity between two microblogs' text (only for text). The TF-IDF formula is:

$$\mathrm{TF}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}. \quad (2)$$

$n_{i,j}$ is the frequency of the particular word in the document *k*, and the denominator is the total number of words in the document. The greater $\mathrm{TF}_{i,j}$ is, the more significant this word is in the document *k*.

$$\mathrm{IDF}_i = \log \frac{|D|}{|\{d : t_i \in d\}|}. \quad (3)$$

$|D|$ is the total number of documents, and $|\{d : t_i \in d\}|$ is the number of the documents that include the word *t*. That means, the greater IDF_i is, the more unusual this word is to all documents.

$$\text{TF-IDF}_{i,j} = \text{TF}_{i,j} \times \text{IDF}_i. \quad (4)$$

Now, from the above equation, we can get the conclusion that: the greater $\text{TF-IDF}_{i,j}$ is, the more representative this word is in the document *k*. Therefore, TF-IDF is a notable measurement to express the similarity between two microblogs' text (only for text).

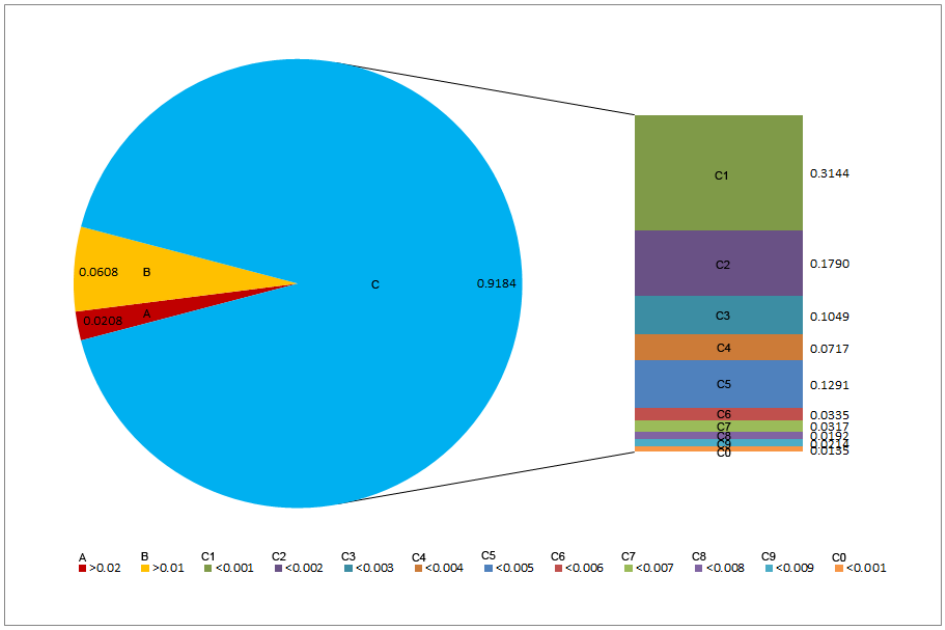

Fig. 2. The distribution of RepSim on the 100 hot microblogs dataset

For example, we have a set of text documents and want to find which document is most relevant to the article "Chinese bee breeding". A simple way to start is eliminating documents that do not contain the three words "Chinese", "bee" and "breeding" at the same time. To further distinguish them, we may count the frequency each term occurs in each document, called Term Frequency (TF), and compare them.

However, because the term "Chinese" is so common, which has appeared too many times in the set, this will tend to incorrectly emphasize documents which happen to use the word "Chinese" more frequently without giving enough weight to the more meaningful terms "bee" and "breeding". The term "Chinese" is not a good keyword to distinguish relevant and non-relevant documents and terms when compared with the less common words "bee" and "breeding". Hence a factor, Inverse Document Frequency (IDF),is proposed, which diminishes the weight of terms that occur too frequently in the documents set while increases those that occur rarely.

So we can see that the TF value increases proportionally to the times a word appears in the document, but the value IDF is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than

others. And TF-IDF is the product of two statistics, term frequency and inverse document frequency, which presents the contribution of a certain word.

Let's focus on the instance of "Chinese bee breeding". Suppose the article has 1000 words, and words "Chinese", "bee", "breeding" all appear 20 times, so the TFs of these words are 0.02. After that, we find 25 billion web pages, and 6.23 billion web pages contain the word "Chinese", 0.0484 billion web pages contain the word "bee", and 0.0973 billion web pages contain the word "breeding". So TF, IDF and TF-IDF are presented in the following sheet:

Table 1. TF, IDF and TF-IDF values of three candidate words

Words	Web pages(bil)	TF	IDF	TF-IDF
Chinese	6.23	0.02	0.60	0.01
Bee	0.05	0.02	2.71	0.05
Breeding	0.10	0.02	2.41	0.09

We can see that the TF-IDF value of "bee" is the highest one. So it is obvious that "bee" is the keyword of the article, which is more representable than other two words.

3.3. Polymerization Degree for Evaluation

The standards mentioned in this section are based on the training set for evaluating the degree of polymerization within the cluster or between the clusters, with the indicators of Cosine, Jaccard and Tanimoto.

Cosine is a simple and popular indicator for evaluating the similarity between vectors. Training data in this paper is vectors showed in Table 2.

$$\text{Cosine}(x, y) = \frac{x^t \cdot y}{\|x\|\|y\|}. \quad (5)$$

The x^t is the transpose of the vector x, and $\|x\|$ is the Euclidean norm of x, and it is the same to $\|y\|$.

The Jaccard index, also known as the Jaccard similarity coefficient, is a statistic used for comparing the similarity and diversity of sample sets, which is defined as the size of the intersection divided by the size of the union of the sample sets.

$$d(i, j) = \frac{r+s}{q+r+s}. \quad (6)$$

$$\text{Jaccard}(i, j) = \frac{q}{q+r+s} = 1 - d(i, j) \quad (7)$$

Where q is the number of vector elements which are not zero at the same time, r and s are the number of vector elements when one is zero and the other is nonzero.

Various forms of functions described as Tanimoto Similarity and Tanimoto Distance occur in the literature and on the Internet. Sometimes Tanimoto is called generalized Jaccard. We calculate the Tanimoto with the formula as following, which is mathematically different from the Jaccard.

$$\text{Tanimoto}(x, y) = \frac{x^t \cdot y}{x^t \cdot x + y^t \cdot y - x^t \cdot y}. \quad (8)$$

Where x^t is the transpose of the vector *x*, and the same to *y*.

We evaluate the degree of polymerization via the following two dimensionalities: within the cluster and between the clusters.

The degree of polymerization within the cluster is calculated based on the similarity formulas described above, containing the item's combination within the cluster. At last, a mean value presents the degree of polymerization within the cluster. We define it as following:

$$\text{Polymerization\_Int}(M) = \frac{\sum_{c \in Combinaton\ \ (M)} Sim}{|Combination\ \ \_Int\,(M)|}. \quad (9)$$

For example, for the cluster {a, b, c} generated by the K-means++ algorithm, the similarities between every two elements in the cluster are as follow:

Sim(a, b) = 0.5, *Sim(b, c)* = 0.6, *Sim(a, c)* = 0.7.

So the polymerization degree within the cluster is:

$$\begin{aligned} \text{Polymerization}_{\text{Int}\,(M)} &= \frac{\sum_{c \in Combinaton\ \ (M)} Sim}{|Combination_{Int\,(M)}|} \\ &= \frac{Sim(a,b) + Sim(b,c) + Sim(a,c)}{C_3^2} \\ &= \frac{0.5 + 0.6 + 0.7}{3} \\ &= 0.6 \end{aligned}$$

The degree of polymerization between clusters is quite similar with the degree within the cluster, except that the combination is between different clusters, rather than within the same cluster. We define it as following:

$$\text{Polymerization\_Ext}(M) = \frac{\sum_{c \in Combinaton\ \ (M)} Sim}{|Combination\ \ \_Ext\,(M)|}. \quad (10)$$

"Int" means "within the cluster", while "Ext" means "between the clusters". These two terms will be used later in this paper.

Finally, we define the polymerization of one time's clustering via formula (11), whose results are used as the global evaluation indicator.

$$\text{Polymerization}(M) = \frac{\text{Polymerization}\ \ \_\text{Int}\,(M)}{\text{Polymerization}\ \ \_\text{Ext}\,(M)}. \quad (11)$$

4. Experiments and Analysis

The experiments are designed as follows to cluster and evaluate Weibo with two indicators, RepSim and TF-IDF. We use K-means++ algorithm to cluster the set of microblogs, and the "distance" in the K-Means++ algorithm are RepSim and the cosine value of TF-IDF vectors. We calculate the TF-IDF value of all the words appearing in

the set. For each microblog, a vector of TF-IDF value of each word appearing in the microblog is available. After that the cosine distance between any two microblogs could be calculated by the vector we got and then K-means++ algorithm runs with the vector, thus our set of microblogs could be separated into K clusters. As for the RepSim, we calculate the RepSim between every two microblogs as the distances in the K-means++ algorithm. Thus, the set of microblogs can also be divided into K clusters.

After clustering, we evaluate the polymerization degree of these two methods. The training data is the standard data for calculating the polymerization degree, and we analyze the statistics at the end of the experiments. First, the polymerization degree between any two pieces of microblogs is computed with the training data vectors, and three kinds of computing methods are Cosine, Jaccard, and Tanimoto. So with the formula of (9), (10), and (11), the polymerization degrees are available, which is important for us to evaluate the results.

4.1. Data Set and Preprocessing

We design a test system to supervise testees to separate microblogs into different classifications in dataset. Seven categories are adopted in our training: Politics, Commerce, Social Focus, ESC (Educational Scientific and Cultural), Sports, Recreation and Health. Testees are well trained and supervised during the whole test process, thus the data training results are credible. We calculate the mean value of all testees' data as our test data:

$$\mathrm{M}_i = \left[m_{i,1}, m_{i,2}, \cdots, m_{i,7}\right]^T$$

$$\mathrm{m}_{i,j} = \frac{\sum_{p \in P} C_{i,j,p}}{|P|}. \tag{12}$$

Where $\mathrm{C}_{\mathrm{i,j,p}}$ is the choice of person *p*,

$$C_{i,j,p} = \begin{cases} 1 & if\ person\ p\ choose\ the\ lable \\ 0 & else \end{cases}. \tag{13}$$

And P is the set of persons who participate in the experiment.

Table 2 shows part of the training results. Since more than one classification options can be selected for a microblog, some sums of the training vector values are greater than 1.

Table 2. Examples of training set

Mid	Politic	Com	Social	ESC	Sport	Recreate	Health
1	0.7	0.2	0.7	0	0	0	0
2	0.4	1.0	0.5	0	0	0	0.2
3	0.2	0	1.0	0	0	0	0.1
4	0.2	0	0.3	0.9	0	0	0
5	0	0	0	0	1.0	0.5	0
6	0	0	0	0	0	1.0	0
7	0	0	0	0	0	0.1	1.0

The detail about training results and the way we select initial centers will be described in the section “Experiments and Analysis” with the Fig. 4 “The distribution of classification after trained”.

The subsequent similarity computing is based on this training set. We now give more introductions about the data training steps:

Firstly, we capture the hot microblogs from Sina Weibo, via the crawler designed by the author through the Weibo Open Platform APIs (http://open.weibo.com). In fact, we have captured over 40,000,000 high-quality users, more than 100,000,000 reposting records, hot microblogs created everyday over one year, and other data. 100 hot microblogs are selected randomly as our test data set.

Fig. 3. The interface of training system

Then, a simple test system is designed to training data, which is like a multiple choice test for testees. We provide good guide to testees for credible results. Fig. 3 gives a screenshot of the training system, where the blue button can submit the classifying results. In this microblog, the text shows poor information for text processing, while the images give people meaningful information.

Finally, statistics about the training results are calculated with formula (12) ~ (13). Some examples are shown in Table 2. Besides, we make a three-dimensional diagram to present both ensemble data and detail of the training data set in Fig. 4.

From the diagram, some information can be found: 1) there are more recreation, social focus and health contents than ESC, commerce, politics and sports; 2) some microblogs have one or more clear classifications, compared to the equivocal; 3) many equivocal microblogs for these seven classifications don’t act well after clustering, mainly because they are extremely confusing on the significance.

That’s the real data from the real SNS site. In general, the 100 top hot microblogs are appropriate to be the test set for Weibo clustering.

Chinese words segmentation is much more difficult than English, especially for the short text. In fact, Chinese short text in SNS (including Weibo) often presents some special features, such as ambiguity and metaphor.

There are several mature and stable open resources for Chinese words segmentation. We refer to these resources and implement a practical program. Specially, artificial detection and modification are made to enhance the accuracy of the TF-IDF based method. The purpose of this operation is to make sure it is more persuasive when compared with our RepSim.

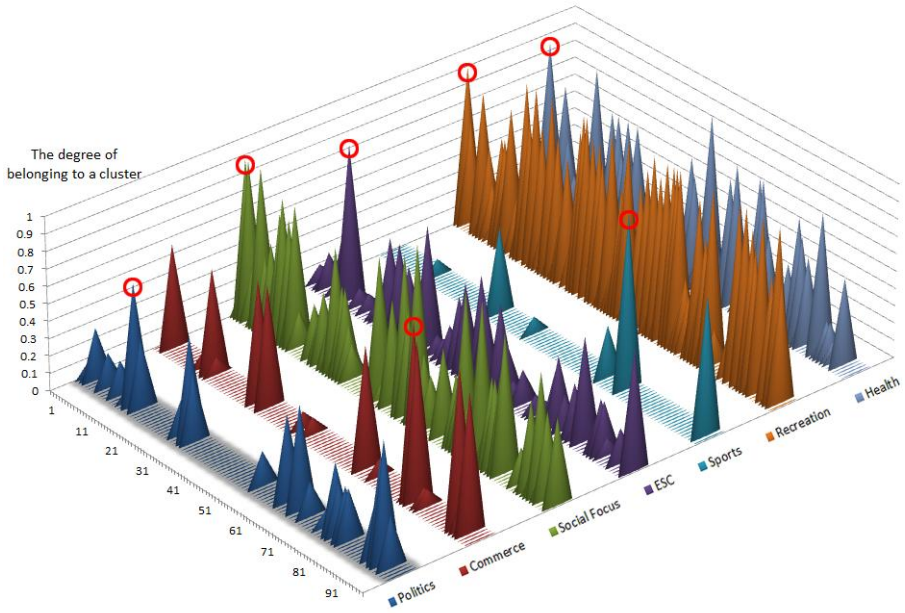

Fig. 4. The distribution of classification after trained

4.2. Clustering and Evaluation

K-means++ are adopted to do the Weibo clustering. The first thing is to select the initial centers. We select the initial centers artificially according to the distribution of classification after trained shown in Fig. 4. Another important thing is the distance computing, here we use RepSim as described and compare with TF-IDF.

The number of clusters is a skillful and experienced job. In this paper, we assume seven classifications, K=7, which is an "ideal" choice. Besides, we set two more options K=3 and K=10 for comparison.

We run the RepSim/TF-IDF based K-means++ algorithms to do the Weibo clustering on the test set, and get 3 sets of results respectively when K=3, 7 and 10. For each results set, Cosine, Jaccard and Tanimoto are calculated. All computing operations are according to formula (5) ~ (11).

Table 3 shows the results of our experiment in detail, from which we can see that:

1) Cosine, Jaccard and Tanimoto act differently but harmoniously. That means, these three indicators play a role in the evaluation and we can get credible analysis results based on them;

2) The degree of polymerization within the cluster is not always greater than the degree between clusters, which seems not so good. But it's acceptable, because that the TF-IDF and RepSim based K-means++ algorithms are simple and not improved specifically. In addition our purpose in this paper is to show the validity of RepSim based K-means++by comparing it with TF-IDF, so whether the RepSim performs better than TF-IDF is much more important to us;

3) In fact, we can find that no matter via the value of Cosine, Jaccard or Tanimoto, the RepSim is better than TF-IDF stably, no matter K=3, 7 or 10.

Table 3. The experiment results. "Int" means the average of indicator in the same cluster's internal; "Ext" means the average of indicator between external clusters

K	TF-IDF						RepSim					
	Cosine		Jaccard		Tanimoto		Cosine		Jaccard		Tanimoto	
	Int	Ext	Int	Ext	Int	Ext	Int	Ext	Int	Ext	Int	Ext
3	0.51	0.51	0.28	0.29	0.40	0.40	0.55	0.49	0.35	0.35	0.36	0.31
7	0.55	0.49	0.28	0.26	0.45	0.38	0.77	0.49	0.66	0.29	0.69	0.34
10	0.58	0.49	0.29	0.30	0.48	0.39	0.72	0.49	0.49	0.26	0.61	0.35

There is a better perspective to make analysis on the evaluation results. With the use of formula (11), we get 18 polymerization values (2 methods (TF-IDF and RepSim) * 3 measurements (Cosine, Jaccard and Tanimoto) * 3 different Ks (K=3, 7 and 10)) at last.

Fig. 5 shows the 18 values with the form of histogram, and a clear contrast can be seen easily with the help of different colors. Especially, the Y-axis presents the polymerization values.

In Fig. 5, we can see that there are 9 pairs containing 2 close neighbors respectively. Take Jaccard (red histogram, while shallow for TF-IDF and deep for RepSim) for example:

1) When K=3, Jaccard based on RepSim is 0.9993, which is 0.62% better than TF-IDF's 0.9931; when K=7, Jaccard based on RepSim is 2.2604, which is 115.44% better than TF-IDF's 1.0492; when K=10, Jaccard based on RepSim is 1.8767, which is 96.27% better than TF-IDF's 0.9561. From the comparison, we can see clearly that the RepSim's global polymerization is better than TF-IDF, especially when K=7 or 10. The results of Cosine and Tanimoto are similar with Jaccard;

2) Another fact is RepSim's polymerizations when K=7 or 10 is always better than K=3 obviously, while TF-IDF's global polymerization is always just so-so and even becoming worse for Jaccard when K=10. That means, TF-IDF is somewhat powerless in Weibo clustering only based on the text, so it performs generally but also "stably". At the same time, RepSim performs much better, and quite robustly;

3) Overall, the entire polymerization when K=7 and 10 is much better than K=3. Especially, the entire polymerization of RepSim based when K=7 is quite conspicuous. This phenomenon reflects that our test training classifies the microblogs into 7 categories. The RepSim based method agrees with the reality well, because it meets the testees' choice.

In conclusion, Fig. 5 indicates that, the RepSim based method is better than TF-IDF, stably and markedly, and new approach utilizing users' reposting data is effective.

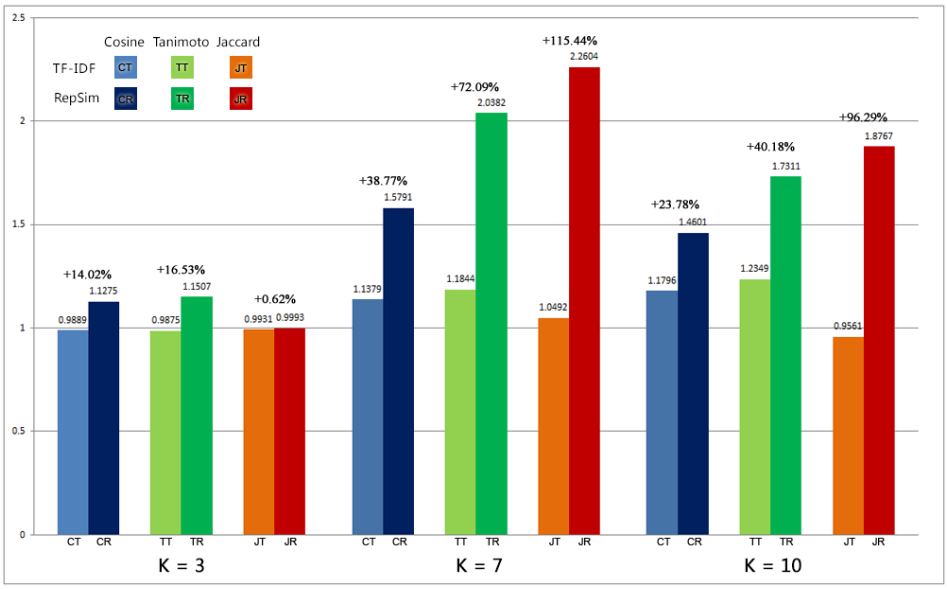

Fig. 5. The comparing between TF-IDF and RepSim via Cosine, Jaccard and Tanimoto when K=3, 7, 10

5. Conclusion

It is a fact that microblogs on the SNS platform are often very short, and text itself only cannot reflect the real interest of the author and the reposting users, so Weibo clustering based on normal methods are not effective any more. Challenges exist in developing novel approaches for Weibo clustering.

Users' reposting behavior data is a good indicator for discovering users' interests. In this paper a new similarity measurement RepSim is proposed for similarity computing between Weibos by analyzing the behavior data of reposting records. Clustering via RepSim is implemented on the hot microblogs from Sina Weibo so as to find the similar topics.

Experiment results indicate that: 1) RepSim performs well on Weibo clustering, especially comparing with the TF-IDF; 2) RepSim is stable and effective in a variety of conditions, including different evaluating standards and K.

There are several advantages about our work. Firstly, RepSim is simple enough to guarantee the real-time performance. Secondly, RepSim depends on the behaviors of users, but few relevant to the contents of microblogs. Considering two microblogs may be similar if they are reposted by the same user. RepSim is born at this moment. Without the interference of the irrelevant contents of microblog, RepSim works better in experiments.

Acknowledgments. This research is sponsored by National Natural Science Foundation of China (61171014, 61272475, 61371185) and the Fundamental Research Funds for the Central Universities（2013NT57）and by SRF for ROCS, SEM. Specially, Libin Jiao and Qin Hu, who are currently undergraduate students in Beijing Normal University, make contribution to this paper in data processing and thesis writing.

References

1. Juan L, Xueguang Z, Bin C.: Research on Analysis and Monitoring of Internet Public Opinion [C]. In Proceedings of the 2012 International Conference of Modern Computer Science and Applications. Springer Berlin Heidelberg, 449-453. (2013)
2. He Z T, Zhang X Q, Zhao F W, et al.: Internet Public Opinion Monitoring Model Based on Cloud Computing [J]. Applied Mechanics and Materials, 404: 744-747. (2013)
3. Wang H.: Understanding Short Texts [M]. Web Technologies and Applications.Springer Berlin Heidelberg, 1-1. (2013)
4. Zhuge H: Schema Theory for Semantic Link Network [J]. Future Generation Computer Systems, Volume 26, Issue 3, March 2010, Pages 408-420. (2010)
5. Sun Y, Bie R, Yu X, Wang S: Semantic Link Networks: Theory, Applications, and Future Trends [J]，Journal of Internet Technology, Vol. 14 No. 3, P.365-378. (2013)
6. D Wilson, DW Supa: Examining Modern Media Relations: An Exploratory Study of the Effect of Twitter on the Public Relations–Journalist Relationship [J]. Public Relations Journal, Vol. 7, No. 3. (2013)
7. Musiał K, Kazienko P.: Social Networks on the Internet [J]. World Wide Web, 16(1): 31-72. (2013)
8. Sun Y, Yan H, Lu C, Bie R, Zhou Z: Constructing the Web of Events from Raw Data in the Web of Things [J], Mobile Information Systems. Volume 10, No. 1, 2014, pp. 105-125. (2014)
9. Jain A K.: Data Clustering: 50 Years Beyond K-means [J]. Pattern Recognition Letters, 31(8): 651-666. (2010)
10. Arthur D, Vassilvitskii S.: K-means++: The Advantages of Careful Seeding[C]. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 1027-1035. (2007)
11. Salton G, McGill M J.: Introduction to Modern Information Retrieval [J]. (1983)
12. Radev D R, Jing H, Styś M, et al.: Centroid-based Summarization of Multiple Documents [J]. Information Processing & Management, 40(6): 919-938. (2004)
13. Lee S, Lee J, Park C Y, et al.: Blog Topic Analysis Using TF Smoothing and LDA [C]. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication.ACM, 75. (2013)
14. Sugiyama K, Hatano K, Yoshikawa M, et al.: Refinement of TF-IDF Schemes for Web Pages Using Their Hyperlinked Neighboring Pages[C]. In Proceedings of the fourteenth ACM conference on Hypertext and hypermedia. ACM, 198-207. (2003)
15. Chum O, Philbin J, Zisserman A.: Near Duplicate Image Detection: Min-Hash and TF-IDF Weighting[C]. In BMVC. 810: 812-815. (2008)
16. Sivic J, ZissermanA.:Video Google: A Text Retrieval Approach to Object Matching in Videos[C]. Computer Vision, 2003.Proceedings.Ninth IEEE International Conference on. IEEE, 1470-1477. (2003)
17. Aizawa A.: An Information-theoretic Perspective of TF-IDF Measures [J]. Information Processing & Management, 39(1): 45-65. (2003)
18. Tang J, Wang X, Gao H, et al.: Enriching Short Text Representation in Microblog for Clustering [J]. Frontiers of Computer Science, 6(1): 88-101. (2012)

19. Li P, Sun Y, Chen Y, Tian Z: Estimating User Influence In Online Social Networks Subject To Information Overload [J], International Journal of Modern Physics B, Vol. 28, No. 3. (2014)
20. Huang M, Yang Y, Zhu X.: Quality-biased Ranking of Short Texts in Microblogging Services [C]. IJCNLP. 373-382. (2011)
21. Sharifi B, Hutton M A, Kalita J K.: Experiments in Microblog Summarization [C]. In Social Computing (SocialCom), IEEE Second International Conference on. IEEE, 49-56. (2010)
22. Liu Z, Yu W, Chen W, et al.: Short Text Feature Selection for Micro-blog Mining [C]. In Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on. IEEE, 1-4. (2010)
23. Phelan O, McCarthy K, Smyth B.: Using Twitter to Recommend Real-time Topical News[C]. In Proceedings of the Third ACM Conference on Recommender Systems. ACM, 385-388. (2009)
24. Efron M.: Hashtag Retrieval in A Microblogging Environment [C]. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 787-788. (2010)
25. Blei D M, Ng A Y, Jordan M I.: Latent Dirichlet Allocation [J]. The Journal of Machine Learning Research, 3: 993-1022. (2003)
26. Ramage D, Dumais S T, Liebling D J.: Characterizing Microblogs with Topic Models [C]. In ICWSM. (2010)
27. Zhang C, Sun J, Ding Y.: Topic Mining for Microblog Based on MB-LDA Model [J]. Journal of Computer Research and Development, 48(10): 1795-1802. (2011)
28. LU Rong, XIANG Liang, LIU Ming-Rong, YANG Qing: Discovering News Topics from Microblogs Based on Hidden Topics Analysis and Text Clustering [J], PR & AI, 25(3): 382-387. (2012)
29. Hu X, Tang L, Liu H.: Enhancing Accessibility of Microblogging Messages Using Semantic Knowledge [C]. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management.ACM, 2465-2468. (2011)
30. Kang J H, Lerman K, Plangprasopchok A.: Analyzing Microblogs with Affinity Propagation [C]. In Proceedings of the First Workshop on Social Media Analytics. ACM, 67-70. (2010)
31. Pal A, Counts S.: Identifying Topical Authorities in Microblogs[C]. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. ACM, 45-54. (2011)

Guangzhi Zhang is currently a postgraduate student in Beijing Normal University, where he is making effort to obtain a Ph.D. degree. He devotes himself to the research of big data and the internet of things.

Yunchuan Sun received his PhD in 2009 from the Institute of Computing Technology, Chinese Academy of Science, Beijing, China. He is currently an associate professor in Beijing Normal University, Beijing, China. He acts as the Secretary of the IEEE Communications Society Technical Subcommittee for the Internet of Things from Jan. 2013. He is also an associate editor of the Springer journal Personal and Ubiquitous Computing. His research interests include Internet of Things, Semantic Link Network, Big Data, Knowledge Representation, Information Security, and Business Models for the Internet of Things. In recent years, he has successfully organized several special issues in some international journals like Springer Personal and Ubiquitous Computing, Elsevier Journal of Networks Computer Applications, etc. He hosts or participates in several research projects from NSFC, 863 Program of China.

Mengling Xu obtained her BSc from Beijing Normal University, and is currently a master graduate student at Beijing Normal University. She has published papers in the area of data mining and semantic link.

Rongfang Bie received her Ph.D. degree in 1996 from Beijing Normal University, where she is now a professor. She visited the Computer Laboratory at the University of Cambridge in 2003. Her current research interests include Internet of Things, Big Data, knowledge representation and acquisition, computational intelligence and model theory.

Received: September 27, 2013; Accepted: March 7, 2014.