Summarizing and Extracting Online Public Opinion from Blog Search Results

Shi Feng[1,2], Daling Wang[1], Ge Yu[1], Binyang Li[2], and Kam-Fai Wong[2]

[1] Northeastern University, Shenyang, China
{fengshi,wangdaling,yuge}@ise.neu.edu.cn
[2] The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China
{sfeng,byli,kfwong}@se.cuhk.edu.hk

Abstract. As more and more people are willing to publish their attitudes and feelings in blogs, how to provide an efficient way to summarize and extract public opinion in blogosphere has become a major concern for both compute science researchers and sociologist. Different from existing literatures on opinion retrieval and summarization, the major issue of online public opinion monitoring is to find out people's typical opinions and their corresponding distributions on the Web. We observe that blog search results could provide a very useful source for topic-coherent and authoritative opinions of the given query word. In this paper, a lexicon based method is proposed to enrich the representation of blog search results and a spectral clustering algorithm is introduced to partition blog search results into opinion groups, which help us to find out opinion distributions on the Web. A mutual reinforcement random walk model is proposed to rank result items and extract key sentiment words simultaneously, which facilitates user to quickly get the typical opinions of a given topic. Extensive experiments with different query words were conducted based on a real world blog search engine and the experiments results verify the efficiency and effectiveness of our proposed model and methods.

1 Introduction

Online public opinion can be defined as the collection of opinions of many different people on the Web and the sum of all their views [14]. Governments have increasingly found public opinion to be useful tools for guiding their public information and propaganda programs and occasionally for helping in the formulation of other kinds of policies. For individual users, public opinion can help them when making decisions.

Nowadays, people are willing to write about their lives and thoughts in blogs, which are often online diaries published and maintained by bloggers, reporting on their daily activities and feelings. The contents of the blogs include commentaries or discussions on a particular subject, ranging from mainstream topics (e.g., food, music, products, politics, etc.), to highly personal interests [5]. According to statistics, there are more than 100 million blogs on the Internet, which has provided us a rich source for extracting public opinion online.

Fig.1 shows the public opinion extraction results in blogosphere for Liu Xiang's withdrawal from Olympic Games [15]. Different from traditional opinion mining

H. Kitagawa et al. (Eds.): DASFAA 2010, Part I, LNCS 5981, pp. 476–490, 2010.

task, the major issue of online public opinion monitoring is to find out the typical opinions and their corresponding distributions on the Web. In Fig.1 there are nine kinds of opinions, and each one reflects a typical point of view toward Liu's withdrawal. For example, about 22% bloggers support Liu Xiang's decision and about 16% bloggers feel disappointed. However, it's tough work for analysts to get this report, because most of the data collecting and typical opinion summarizing tasks can only be done manually.

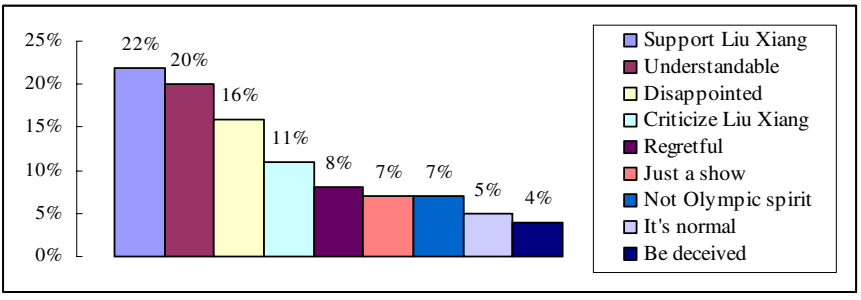


Fig. 1. Online public opinion for Liu Xiang's withdrawing from Beijing Olympic Games

From the discussion above we know that online public opinion extraction is not just a sentiment classification task or the same as opinion summarization task. The three major challenges include:

How to sample the blogosphere. Since there are huge amount of blogs on the Web, given a topic, it is unrealistic to analyze all the topic relevant blogs in the blogosphere. A sampling strategy need to be designed so that we can use a small dataset which could represent as many bloggers' opinions as possible.

How to find the typical opinions. A typical opinion is a point of view held by many people. Public opinion monitoring in blogosphere should aggregate individual attitudes or beliefs and extract typical opinions in the sample dataset.

How to quantitatively measure the distribution of typical opinions. As the example in Fig.1, we should know how many people "*support Liu Xiang*", and how many people "*feel disappointed*" in the dataset, so that we can get a macro view of people's attitudes toward the given topic on the Web.

Most recently, opinion mining techniques have been used to find people's attitudes in blogosphere. Previous studies on opinion retrieval in blogs usually focus on finding the topic relevant opinionated blog entries [22][23], but not the opinion relatedness between the retrieval results. The existing studies on opinion summarization can generate a short abstract of the major opinions in a close blog dataset on a given topic [6]. However, for public opinion monitoring task, the extracted results should not only contain the summary of opinions, but also should include the distribution of each typical opinions. Moreover, previous studies on opinion retrieval and summarization equally treat each blog document, but in real world, blogs from opinion leaders, who

have a greater influence on the Web, should be given priority during the summarization. Therefore, there are still some defects in existing methods, which could not totally meet the need for public opinion monitoring task in blogosphere.

In this paper, we propose a new method to summarize and extract public opinion from blog search results. We use the titles and snippets of **b**log **s**earch **r**esults (BSRs for short) to summarize public opinion based on the following considerations:

(1) In many cases, bloggers do not confine themselves to one topic in a blog article. But sophisticated Web search techniques could guarantee that BSRs are highly topic relevant to the query word;
(2) Usually, these titles and snippets in BSRs contain bloggers' opinion about the given query word;
(3) Due to the algorithms of blog search engines, the top ranked BSRs are from popular or opinion leader's blog sites. So we can get the public opinion of the whole blogosphere using a relative small BSRs dataset.
(4) We need not to crawl and index the huge amount of blog entries on the Web.

In order to tackle the above three challenges, several hundreds of the top ranked BSRs are used to sample the blogosphere about the given topic. Then a lexicon based method is proposed to measure the underlying opinion relatedness between BSR items and a spectral clustering method is employed to aggregate the opinions into groups, which reflect the opinion distributions on the Web. Finally, a mutual reinforcement random walk model is proposed to rank BSRs and extract key sentiment words in each opinion cluster, which facilitates user to quickly get the typical opinions of the given topic in blogosphere.

To our best knowledge, this is the first paper trying to summarize and extract online public opinion from blog search results. The rest of the paper is organized as follows. Section 2 analyzes the sentiment characteristics of BSR items and discusses the new sentiment representation for BSRs. Section 3 describes opinion clustering algorithm. In Section 4, we will propose a random walk model to rank BSR items and extract key sentiment words simultaneously. Section 5 provides experimental results on real world blog search engine. Section 6 introduces the related work. Finally we present concluding remarks and future work in Section 7.

2 Blog Search Results Sentiment Representation

In this section, we attempt to give BSRs a new representation in order to measure the opinion relatedness between BSR items.

2.1 Characteristics of Blog Search Results

Some commercial blog search engines have been published on the Web [4] [16]. These services usually use Web search techniques and rank the results by their topic relevance and the popularity of the blog entry. To demonstrate the sentiment characteristics of BSRs, we issue the movie name "*Hancock*" in Google Blog Search, then several results are collected and shown in Table 1.

Table 1. The titles and snippets for query word "*Hancock*"

Title	Snippet
Hancock	Go see Hancock. Much respect to Will Smith and the directors behind the film, truly inspirational.
It's a comedy, It's a fantasy, Yes, It's 'Hancock'	Hancock was enjoyable, but no without it problems thanks to many unanswered questions.
Hancock 2008 DVDRip direct links	Hancock 2008 Language: English Runtime: 92 min Country: USA Release Date: 2 July 2008.

We can see that the BSR items in Table 1 have the following characteristics:

(1) The title and snippet of each item are very short, may be just one or two sentences, and sometimes may be just a word.

(2) The titles and snippets are highly relevant to the given query word. That's because the blog search engine employs sophisticated and mature Web search techniques to get the most topic relevant articles and snippets;

(3) Some of the titles and snippets contain the bloggers' sentiments and opinions. As the search results are highly topic-coherent, the sentiment words in titles and snippets mainly reflect the bloggers' own opinions about the given query key word;

(4) Not all the blogs contain authors' emotions, there are some informative results mixed up with affective ones. For example, the last result item in Table 1 tells us the download information of Hancock movie DVDRip;

According to [12], there are two kinds of blog articles in the blogosphere, namely informative blogs and affective blogs, and Table 1 also confirms this point of view. Here we give our definition of Affective BSR and Non-affective BSR.

Affective BSR. An Affective BSR is the BSR item that contains bloggers' sentiments and opinions.

Non-affective BSR. The contents of this genre of BSR include (1) the informative BSR that providing or conveying information and (2) short snippet that do not contain any personal feelings and emotions.

An opinion usually includes opinion holder, opinion target and sentiments. According to the properties of blogs, the opinion holder of a BSR item is the blogger himself/herself. The opinion target is usually the query word or the subtopic related to the query word. From this observation, we submit topic words to a blog search engine, collect the BSR items and aggregate the sentiments in BSRs, so as to summarize and extract the public opinion of the bloggers about the given topic.

2.2 BSRs Sentiment Representation

BSRs are usually very short and opinion words usually do not converge like topic words. So directly applying traditional similarity measure based on term matching to BSRs often produces inadequate results. In this section, we propose a new sentiment representation for BSRs based on WordNet gloss.

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct

concept [19]. Each synset in WordNet has a gloss that defines the concept that it represents. The intuition of this paper is that the terms with similar sentiments have similar glosses. For example, the synset **A** contains the words "*amusing*", "*amusive*", "*diverting*" and "*fun*" and it has the gloss "*providing enjoyment; pleasantly entertaining*"; Synset **B** contains the words "*amused*", "*diverted*", "*entertained*" and it has the gloss "*pleasantly occupied*". The words in synset **A** and **B** are quite different. However, their glosses share the same word "*pleasantly*". Therefore, synset **A** and **B** have similar sentiment meanings, i.e. when people use the words in **A** and **B**, they tend to express similar state of emotions. In this paper, we attempt to remove non-sentiment words and add the glosses of each emotion-bearing word into BSRs to give them a new sentiment representation. The details of each step are discussed as follows:

Step 1. Lemmatization. We convert the words into their basic lemma form. We do not conduct stemming algorithm to the words because we must keep their original sentiment meanings.

Step 2. Negation Processing. Each word in the negation sentences is replaced by its antonym in WordNet and the words that do not have antonyms in WordNet are given a new prefix "*not-*".

Step 3. Sentiment Words Tagging. In this paper, we use SentiWordNet as the sentiment lexicon. Extensive experiments show that SentiWordNet is a very effective lexicon tool for finding emotion-bearing words [1][7]. Words with positive or negative strength above a threshold in SentiWordNet are picked out and corresponding words in BSRs are tagged as sentiment words.

According to definition in Section 2.1, suppose R_a represents the set of Affective BSR and R_{na} represents the set of Non-affective BSR. So we get $\text{BSRs} = R_a \cup R_{na}$. Let *sw* denote sentiment word and r, r_i, $r_j \in$ BSRs. If r contains sentiment word, we say $sw \in r$, so we employ the following way to classify BSRs:

$$R_a = \{r_i \mid (\exists sw, sw \in r_i)\}, R_{na} = \{r_j \mid (\forall sw, sw \notin r_j)\} \quad (1)$$

The above formulas indicate that if r contains at least one sentiment word, we classify it into Affective BSR category; otherwise, we classify it into Non-affective category. It must be emphasized that since our goal is to group BSRs by their opinions, we do not care about the sentiment orientations of each word in BSRs. After this step, we eliminate search result items in R_{na}, and the words in R_a that don't have sentiment tags are also removed. Only sentiment words in R_a are brought to the next processing steps.

Step 4. BSRs Sentiment Representation. After prior processing steps, each remaining BSR item can be represented by a set of sentiment words. Give a BSR item r containing n sentiment words, we have $r = \{sw_1, sw_2, \ldots, sw_n\}$. Synsets and glosses in WordNet are used to expand sentiment words representations, so we have $E(sw) = \{Synset(sw), Gloss(sw)\}$. $Synset(sw)$ denotes all words in the synset of sw; $Gloss(sw)$ denotes the gloss of sw. Therefore, we expand a BSR item r as $Er = \{E(sw_1), E(sw_2), \ldots, E(sw_n)\}$.

We concatenate the expansions of sentiment words in r together, and vector space model is used to represent BSRs. Suppose r_j is a BSR item:

$$\vec{r}_j = (TFIRF(t_1), TFIRF(t_2), ..., TFIRF(t_m)) \quad (2)$$

where t_i (i=1, 2, …, m) is a term in the new representation, i.e. $t_i \in Er$, TF represents term frequency in the BSRs and IRF denotes the inverse BSR item frequency. We call the new representation of BSRs as Sentiment Vectors (SV), because it can reflect bloggers' original emotions and opinions.

3 Aggregate Opinions in BSRs Based on Spectral Clustering

Blogger's opinions about a certain topic may be opposite or quite different. Our intention is to not only summarize typical opinions, but also find out their corresponding distributions, i.e. how many people hold similar opinions in the blogosphere. So in this section we attempt to aggregate BSRs into opinion clusters.

3.1 Sentiment Similarity Computing

In Section 2.2, we introduce SV to represent bloggers' emotions in BSRs. SV have enriched the representations of the emotion-bearing words in each BSR item, and we employ traditional text similarity measurement algorithm to compute the sentiment similarity between BSR items. Evaluating a variety of similarity measurement algorithms on SV, however, is not the aim of this paper. Rather we simply want to find out whether the new expanded representations of BSRs are effective in reflecting bloggers' opinions. In this paper, we consider BSR items as nodes, and the BSRs collection can be modeled as an affinity graph in which each link denotes the sentiment similarity between BSR items. Formally, given the BSRs set R, let G=(V,ED) be an affinity graph to reflect the relations between items in R. V is the set of vertices and each vertex $v \in V$ is an item in the BSRs set. ED is the set of edges. Each candidate edge e_{ij} in ED is associated with a similarity weight between item v_i and v_j. The similarity weight function $SentiSim$ is defined as:

$$SentiSim(\vec{v}_i, \vec{v}_j) = \frac{\vec{v}_i \cdot \vec{v}_j}{\| \vec{v}_i \| \times | \vec{v}_j |} \quad (3)$$

We use adjacency matrix M to describe the structure of G and the value of M_{ij} represents the weight of an edge in the graph. So we have

$$M_{ij} = \begin{cases} SentiSim(\vec{v}_i, \vec{v}_j), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Our intention is to aggregate similar opinions in BSRs, namely we have to partition the graph G into several subgraphs and each subgraph should reflect coherent opinions of the bloggers. This is not an easy task because we do not know the structure of the graph G. Moreover, the number of clusters could not be easily predicted in advance. In the next section, we employ a spectral clustering algorithm to partition graph G which does not need to make any assumptions on the form of the clusters and a heuristic method to determine the number of clusters is introduced.

3.2 Spectral Clustering for BSRs

Spectral clustering is an effective algorithm based on graph partitioning. The basic idea of the algorithm is to map the raw data space into eigenspace. In this paper, we choose the MS [8] method with the computation of the Laplacian matrix as follows. Let D be the diagonal degree matrix of M, i.e. $D_{ii} = \sum_j M_{ij}$. The Laplacian matrix L is defined as $L = I\text{-}D^{-1}M$. The first k generalized eigenvectors of L is found to compose a new matrix M' and the traditional clustering method such as K-Means can be used on M' to find clusters.

The number of clusters. We could not know the cluster number k in advance. In this paper, we employ a heuristic algorithm to auto determine k by computing eigengap of the matrix L. Matrix perturbation theory indicates that the stability of the eigenvectors of a matrix is determined by the eigengap. However, sometimes the cluster structure of the data is not so obvious, or there may be several big eigengap candidates, i.e. there are several eigenvalues λ_k where $|\lambda_{k+1}\text{-}\lambda_k|$ is large. So we use candidate eigengaps to heuristically set the value of k and evaluate the quality of the clustering results to get the best k. Based on the assumption that the best partitioning will have most edges within the subgraphs and little edges between subgraphs, the quality of graph partitioning is defined as [11]:

$$Q(C) = \sum_{i=1}^{k} (e_{ii}/c - (a_i/c)^2) \tag{5}$$

where C is a candidate clustering result, k represents the number of clusters, e_{ii} is the number of edges with both vertices within cluster i, a_i is the number of edges with one or both nodes in cluster i, and c is the total number of edges. The heuristic method to determine k is as follows: (1) Compute the eigenvalues of L; (2) Find the biggest three eigengaps, and set the candidate number of clusters k_1, k_2, k_3; (3) For each candidate k, we employ the MS spectral clustering method [8] to partition G into subgraphs; (4) The $Q(C)$ function in Formula 5 is used to evaluate each candidate clustering results. The best $Q(C)$ is chosen, so the final clustering result and the number of cluster are confirmed.

We call this opinion clustering algorithm as OC algorithm. Using OC algorithm, we can generate opinion coherent clusters of a given BSRs dataset. At the same time, we hope that each BSR item could be ranked by its sentiment coherence to the semantic meanings of the cluster. We will discuss this opinion ranking and keywords extraction method in the next section.

4 Opinion Ranking and Keywords Extraction

When browsing the Web search results, people used to read several top ranked items. Based on this intuition, the proposed algorithm should not only group BSRs into opinion clusters, but also should rank the results in each cluster according to certain metric. Considering the intention of people exploring the blogosphere, the words with higher sentiment strength are better indicator for bloggers' emotions and the BSRs contain definite sentiment orientation and strong emotion meanings will attract more

attention. Therefore, the proposed algorithm should rank these BSRs in higher position. And we also hope that the key sentiment words are extracted for each cluster, which could facilitate users' quick browsing through the public opinion summarization results.

Inspired by the work of Wan [18], in this paper we propose a mutual reinforcement random walk model to rank BSRs and extract key sentiment words simultaneously. Our basic assumption is that a BSR item is important if it includes important sentiment words and is heavily linked with other important BSR items. And also, a sentiment word is important if it has higher sentiment strength; it appears in many important BSR items and has relation with many other important sentiment words. This mutual reinforcement relationship of BSRs and sentiment words is shown in Fig.2.

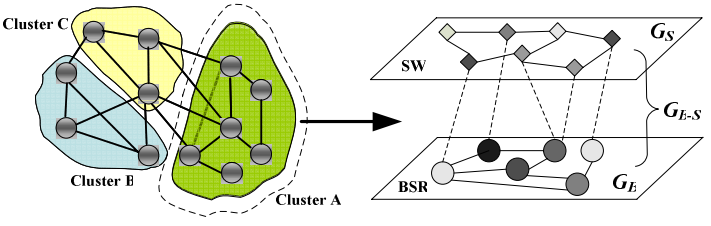


Fig. 2. The mutual reinforcement of BSRs and sentiment words

In Fig.2, given a BSRs dataset, OC algorithm has partitioned the graph into several clusters. In a cluster, BSRs represents the blog search results in a cluster; SW denotes the sentiment word set of the given cluster. We build three graphs G_B, G_S and $G_{B\text{-}S}$ to reflect the BSR-BSR, SW-SW, BSR-SW relationship. For bipartite $G_{B\text{-}S}$ graph, if a sentiment word sw_j appears in BSR r_i, an edge will be created between r_i and sw_j. Each node in these graphs is associated with a sentiment strength value (shown as different grayscale in the right part of Fig. 2), and based on the random walk on these graphs, this strength is diffused in the three graphs. After several mutual reinforcement iteration steps, the important BSR items could be ranked in higher position and simultaneously we also get the most salient sentiment words in each cluster. The detail of the algorithm is described as follows.

Given a cluster C_o that is the subgraph of G, we have the new adjacency matrix $\{B \mid B_{ij} = M_{ij} \text{ where } i, j \in C_o\}$ to represent G_B. We use S to denote the adjacency matrix of G_S and the similarity between SW is calculated by cosine similarity of WordNet expansion representation $E(sw)$. The adjacency matrix of BSR-SW relationship is represented by W, and the weight is computed as:

$$W_{ij} = \frac{TFIRF(sw_j)}{\sum_{sw \in r_i} TFIRF(sw)} \tag{6}$$

where given a sentiment word sw_j in BSR item r_i. If sw_j appears frequently in r_i and seldom appears in other BSR item, there is a higher weight between sw_j and r_i.

B, S and W is normalized to $\tilde{B}$, $\tilde{S}$ and $\tilde{W}$ respectively and the normalized transpose of W is represented by $\hat{W}$. Let R_{BSR}, R_{SW} denote the ranking scores of BSR and SW. The mutual reinforcement random walk approach can be formulated as follows:

$$\begin{cases} R_{BSR}^{(k+1)} = \alpha \tilde{B}^{\mathrm{T}} R_{BSR}^{(k)} + (1-\alpha)\hat{W}^{\mathrm{T}} R_{SW}^{(k)} \\ R_{SW}^{(k+1)} = \beta \tilde{W}^{\mathrm{T}} R_{BSR}^{(k)} + (1-\beta)\tilde{S}^{\mathrm{T}} R_{SW}^{(k)} \end{cases} \tag{7}$$

Suppose we have:

$$Y = \begin{bmatrix} \alpha \tilde{B}^{\mathrm{T}} & (1-\alpha)\hat{W}^{\mathrm{T}} \\ \beta \tilde{W}^{\mathrm{T}} & (1-\beta)\tilde{S}^{\mathrm{T}} \end{bmatrix}, \quad R = \begin{bmatrix} R_{BSR} \\ R_{SW} \end{bmatrix} \tag{8}$$

In matrix form, we have the equation $YR=\lambda R$. Similar to the idea of PageRank [13], we add links from one node to any other nodes in G_B and G_S graph, so we have:

$$Y = \begin{bmatrix} \alpha((1-d)E/n_1 + d\tilde{B}^{\mathrm{T}}) & (1-\alpha)\hat{W}^{\mathrm{T}} \\ \beta \tilde{W}^{\mathrm{T}} & (1-\beta)((1-d)E/n_2 + d\tilde{S}^{\mathrm{T}}) \end{bmatrix} \tag{9}$$

where E is a square matrix with each element equal 1. We can prove that the transpose of Y is stochastic and irreducible.

Lemma: Y^{T} is irreducible and when $\alpha + \beta = 1$, it is stochastic.

Proof: There is a link between each node in G_B and G_S, so they are strong connected. Because $G_{B\text{-}S}$ has connected the nodes in G_B and G_S graph, for each pair of nodes u, v in these three graphs, there is a path from u to v. Therefore, the new graph G_{All} composed by G_B, G_S, $G_{B\text{-}S}$ is strong connected. And also there will be more than one path for any pair of nodes in G_{All}, so G_{All} is aperiodic and the matrix Y^{T} is irreducible. For any column in the left part of Y:

$$\sum\nolimits_i Y_{ij} = \alpha\left(\sum_{i=1}^{n_1} \frac{1-d}{n_1} + d\sum_{i=1}^{n_1} \tilde{B}_{ij}\right) + \beta\sum_{i=1}^{n_3} \tilde{W}_{ij} = \alpha + \beta \tag{10}$$

The same conclusion can be deduced in the right part of Y. So when $\alpha + \beta = 1$, the sum of elements in each column in Y is 1, and the matrix Y^{T} is stochastic. □

E/n in Formula 9 means that each node in the graph has an equal weight. Recall our assumption that the word with higher sentiment strength is a better indicator for bloggers' emotions, we give each node a different weight during the iteration steps. For graph G_S, the weight of a node is defined by the sentiment strength of the word sw, and we have the weight vector $\{q|q_i = f(sw)\}$, where $f(sw)$ is the sw's sentiment strength in SentiWordNet. For graph G_B, the weight is defined as the average strength of the word in each BSR item, and we have $\{p \mid p_i = \sum_{j=1}^{N} f(sw_j) \Big/ N\}$. p and q is normalized to $\tilde{p}$ and $\tilde{q}$. The matrix Y can be reformulated as follows:

$$Y' = \begin{bmatrix} \alpha((1-d)e\tilde{p}^{\mathrm{T}} + d\tilde{B}^{\mathrm{T}}) & (1-\alpha)\hat{W}^{\mathrm{T}} \\ \beta \tilde{W}^{\mathrm{T}} & (1-\beta)((1-d)e\tilde{q}^{\mathrm{T}} + d\tilde{S}^{\mathrm{T}}) \end{bmatrix} \tag{11}$$

Using Formula 11, we incorporate sentiment strength information into mutual reinforcement random walk model and it can be proved that Y'^{T} is stochastic and irreducible. The power method is used to iteratively find the solution of the equation $Y'R=R$. It is guaranteed that R will converge to a steady state, which we use as final ranking results of BSR items. Finally, the top 5 ranked sentiment words are extracted as key sentiment words of the cluster. This ranking and summarization method is applied on each cluster.

5 Experiments

5.1 Experiment Setup

Our experiment is conducted on a commodity PC with Windows XP, Core2 Duo CPU and 4GB RAM. Given a query key word, we use Google Blog Search to find the topic relevant blog entries. Titles and snippets are parsed and extracted for further processing steps.

Web search results clustering usually focus on informative, polysemous and poor query words, such as "*java*", "*jaguar*", "*apple*". With different purpose, we pay more attention to the entities' and events' name which can arouse people's interest to publish opinions in blogosphere. The different types of query words used in this paper are shown in Table 2.

Table 2. The query words used in the experiments

ID	Type	Query Words	Data Range
Hancock	Movie	hancock movie	2008.7.1-2008.7.31
Obama	People	president obama	2009.1.1-2009.1.31
Opening	Event	beijing olympic opening	2008.8.8-2008.8.12
IPod	Product	ipod touch	2007.9.1-2007.9.31

Public opinions are highly relevant to the published time. If there is a hot topic emerging on the Internet, people are eager to write down their own opinions about the topic in blogs. However, as the time passing by, the blogs on original topic become fewer and fewer and there maybe a succeeding topic or a new story emerging in the blogosphere. In this paper, we restrict the publishing date of blogs for the query to find the most topic coherent story in the searching space. For example, we restrict the query "*hancock movie*" within one month period since the movie was released.

Usually blog search engines have already ranked results by blogs' authorities, and opinion leaders' blogs and popular blogs can be returned in the higher position. Therefore, we use a relatively small dataset to reflect the major opinions in the whole blogosphere. As a result, less than 1000 BSR items are collected for each query.

Evaluation Measure. There is no ground truth for the clustering results. Since we have model BSRs as graph, the best partitioning results would have most edges with the cluster and little edges between the clusters. So we use Formula 5 in Section 3.2 to evaluate clustering performance.

We use precision (*P*) at top *N* results to measure the performance of the ranking algorithm. Since no golden standard is available for these search results, we ask three graduate students major in opinion mining to evaluate the opinion coherence between ranked search results and extracted key sentiment words in each cluster. However, it is very subjective to measure this opinion coherence. The three evaluators are asked to browse through the search results, compare key sentiment words to each result and give a score ranging from 0 to 10 to show how well the extracted key sentiment words match the opinion contained in each search result. If the score is high, it means that the extracted words are good opinion summarization for the given search result item. For each cluster we have:

$$p@N = \frac{\sum_{i=1}^{N} Score_i / 10}{N} \tag{12}$$

where *N* denotes the top *N* result items in the cluster, and $Score_i$ is the value given by evaluators. Notice that $p@N$ represents the precision of one cluster, and the average precision of all clusters is calculated as $P@N = (\sum_k p@N)/k$. Finally, we use the average $P@N$ value of the three evaluators to measure the performance of proposed BSRs public opinion summarization and extraction algorithm.

Using Sentiment Vectors and OC algorithm, we hope that most of the Non-affective BSR items are filtered out and the affective ones are ranked in high position. Thus we use Non-affective BSR (*I*) at top *N* results to measure this performance:

$$I@N = \frac{I \cap N}{N} \tag{13}$$

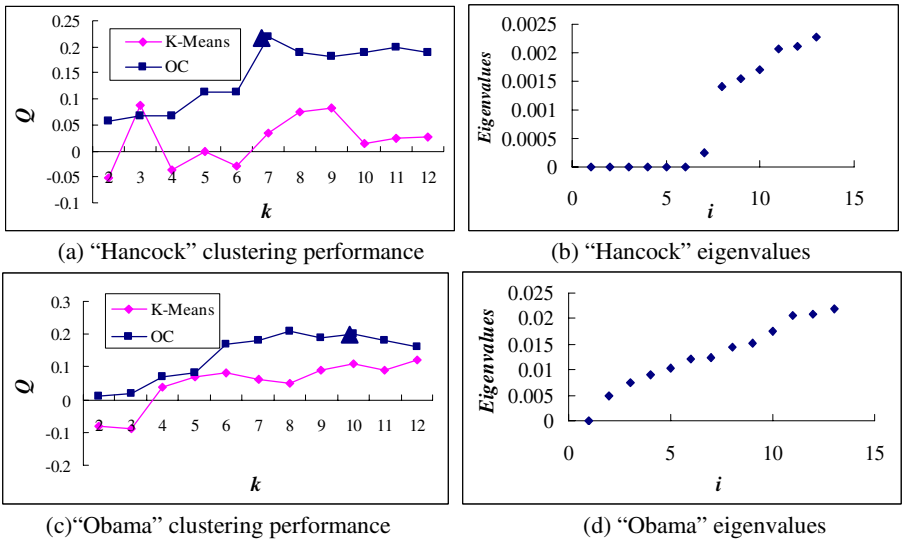
where *I* is the number of Non-affective BSR items in top *N* results. Three human annotators are asked to label BSR item with A/NA tag, i.e. Affective/Non-affective result. If there is a disagreement between the first two annotators, the third one will decide the final tag of the BSR item.

5.2 Experiment Results

Opinion Clustering Performance. We compare the proposed opinion clustering method with the basic K-Means method. To validate the effectiveness of the cluster number determination algorithm, we manually set *k* from 2 to 12.

Fig.3 (a) and (b) show the clustering performance and eigenvalues of the query "*hancock movie*". In Fig.3 (a), the Y axis denotes the $Q(C)$ function value. The bigger triangle represents the auto-determined cluster number of OC algorithm. It can be seen from Fig.3 (b) that the "Hancock" dataset has a very obvious eigengap, and the proposed OC algorithm could find the best clustering performance using this eigengap when k=7. The "Beijing Olympic opening" and "IPod Touch" dataset also validate the proposed method (due to space limitation, we do not list the figures here). However, in Fig.3 (c) OC algorithm does not find the best partition number (OC predicts k=10, but the best Q is achieved when k=8). Fig.3 (d) illustrates that the difference between all eigenvalues are approximately the same for the "President Obama" dataset. This indicates that there is no clear cluster structure in "Obama" dataset, and this may

because the sentiment words used in this dataset are quite scattered. In this situation, we hope that our ranking algorithm could help us to figure out typical opinions in BSR items.



(a) "Hancock" clustering performance (b) "Hancock" eigenvalues

(c)"Obama" clustering performance (d) "Obama" eigenvalues

Fig. 3. Clustering performance results

Opinion Ranking Precisions. Here we compare the proposed mutual reinforcement random walk based opinion ranking algorithm (MR algorithm) with two different ranking methods. The first method is directly rank BSR items by average sentiment strength, and we call it SR. The second one is directly employ basic PageRank algorithm on the BSR graph, which does not consider sentiment strength and we call it PR. The key sentiment words of these two methods are extracted by *TFIRF* function, i.e. the words with top *TFIRF* values are extracted as key sentiment words. Here we set $\alpha=\beta=0.5$ of MR algorithm to equally treat the weights of BSRs and sentiment words. The parameter d is set to be 0.85. We use MR-NE represents the mutual reinforcement algorithm without WordNet gloss expansion, i.e. the similarity is computed directly based on sentiment words vectors. The comparison of opinion ranking performance is shown in Fig.4.

We can see from Fig.4 that generally the proposed MR method outperform the other method. Note that the precision is calculated by volunteers' comparing ranked BSRs with extracted key sentiment words. If the key sentiment words could reflect the major opinions expressed in the BSRs, a higher score will be given. Fig.4 validates that the mutual reinforcement method could effectively rank opinions and extract key sentiment words for each cluster and the extracted words could provide a very brief summarization of the major opinions in each cluster. And also we can conclude that the WordNet synset and gloss expansion are effective in finding the underlying opinion relatedness between short BSR texts.

The best performance is achieved using the query "*beijing olympic opening*". After analyzing the search results, we find that the words that people used to express their

opinions on Beijing Olympic open ceremony are really converging. On the other hand, the words reflecting people's opinions on political figures are more complex and scattered. Thus the precision of our algorithm is decreased.

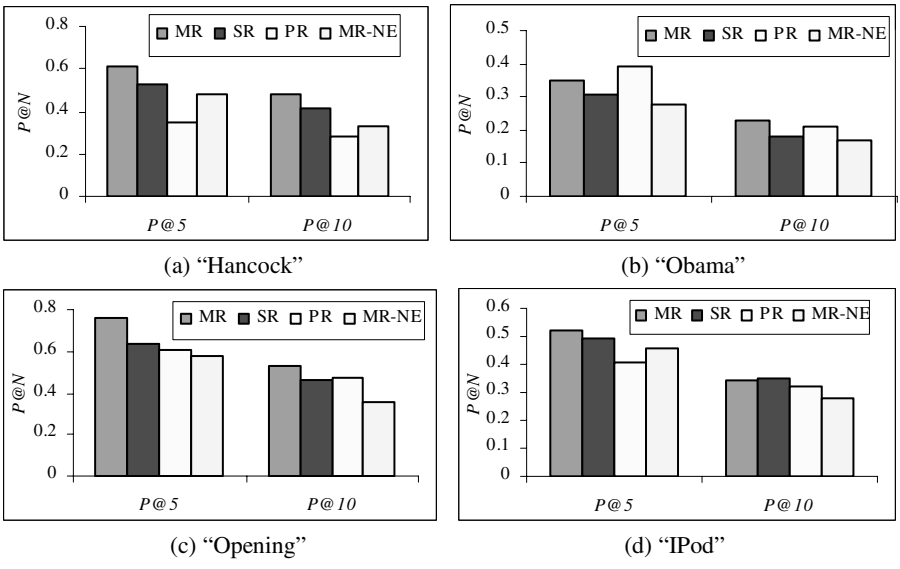


Fig. 4. Opinion ranking performance

Affective vs Non-affective. The *I@N* performance using query "*hancock movie*" and "*president obama*" is shown in Fig. 5. It can be seen from Fig. 5 that *I@5* is relatively small. Using opinion ranking algorithm, there is average 0.6 items in the top 5 BSRs are non-affective for query "*hancock movie*", compared with average 4.6 non-affective items in top 15 BSRs. Therefore, it can be concluded that the proposed algorithm can effectively rank the affective BSR in high position.

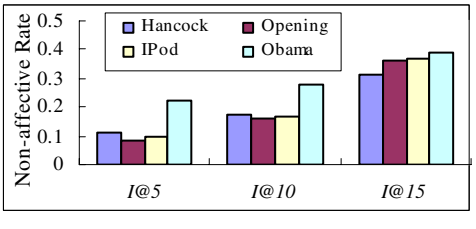


Fig. 5. Performance of *I@N*

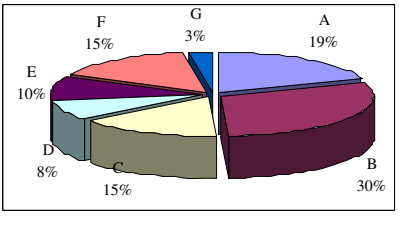


Fig. 6. Opinion distribution for "*hancock movie*"

Fig. 6 illustrates the clustering results for the query "*hancock movie*". 924 BSR items are parsed for the further clustering steps. Key sentiment words are extracted for each cluster. Take cluster A for example, the extracted words include "*good*" and "*pretty*" and the top rank BSR items contain the sentences such as "*I think overall is*

good" and "*saw Hancock yesterday and that was pretty good*". Generally, the proposed method can extract sentiment words which reflect the major opinions expressed in the top ranked results. However, human evaluation for these results is very subjective. The new evaluation method without human involvement will be presented in the future work.

6 Related Work

6.1 Search Results Clustering

Determining the similarity of short text snippets works poorly with traditional document similarity measures. Some lexicon-based and language modeling-based have been proposed to solve the text snippets similarity measurement and clustering problem [9] [20]. But there are still some obstacles to measure the sentiment similarity between the short texts.

Rich literatures have been published on search results clustering. Zeng et al. [21] reformalize the search result clustering problem as a supervised salient phrase ranking problem. Ferragina et al. [3] develop an open-source system which offers both hierarchical clustering and folder labeling with variable-length sentences. There are already some industrial Web search results clustering systems on the Internet [17] [10], which are especially useful for informative, polysemous and poor queries.

We have proposed a sentiment clustering method for blog search results [2]. However, in [2] the sentiment similarity between BSRs is only considered at word level and key words are extracted only by sentiment strength.

6.2 Opinion Mining

The task of opinion retrieval is to find relevant and opinionate documents according to a user's query [22] [23]. TREC started a special track on blog data in 2006 with a main task of retrieving personal opinions towards various topics, and it has been the track that has the most participants in 2007 [7]. However, people could have various opinions on the same topic, and opinion retrieval can not provide users with overall summarization of opinions expressed in blog articles.

TAC 2008 has launched a task on opinion question answering and summarization. Given a list of questions, an exact string answer or several sentences containing the answer should be returned [6]. Different from that task, our intention is to find bloggers' typical opinions and their corresponding distribution in blogosphere.

7 Conclusion and Future Work

Blog has provided a good platform for people to express their opinions and attitudes. In this paper, we propose a method to summarize and extract public opinion based on blog search results. Opinions are aggregated into clusters. A mutual reinforcement random walk model is proposed to rank blog search result items and extract key sentiment words. Experimental results demonstrate that the proposed method can effectively extract public opinion in the blogosphere.

In this study, only sentiment words are used to represent bloggers' opinions in BSRs. In future work, more linguistic information may be considered in the new representation of sentiment vectors.

Acknowledgments. This work is partially supported by National Natural Science Foundation of China (No.60973019, 60973021), CUHK Direct Grant Scheme (No. 2050443, 2050417) and HKSAR's ITF (No. ITS/182/08).

References

1. Esuli, A., Sebastiani, F.: SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: Proceedings of LREC, pp. 417–422 (2006)
2. Feng, S., Wang, D., Yu, G., Yang, C., Yang, N.: Sentiment Clustering: A Novel Method to Explore in the Blogosphere. In: Li, Q., Feng, L., Pei, J., Wang, S.X., Zhou, X., Zhu, Q.-M. (eds.) APWeb/WAIM 2009. LNCS, vol. 5446, pp. 332–344. Springer, Heidelberg (2009)
3. Ferragina, P., Gulli, A.: A Personalized Search Engine based on Web-snippet Hierarchical Clustering. In: Proceedings of WWW, pp. 801–810 (2005)
4. Google Blog Search, http://blogsearch.google.com
5. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: Structure and Evolution of Blogspace. Commun. ACM 47(12), 35–39 (2004)
6. Li, F., Tang, Y., Huang, M., Zhu, X.: Answering Opinion Questions with Random Walks on Graphs. In: Proceedings of ACL, pp. 737–745 (2009)
7. Macdonald, C., Ounis, I., Soboro, I.: Overview of the TREC-2007 blog track. In: Proceedings of TREC 2007 (2007)
8. Meila, M., Shi, J.: Learning Segmentation by Random Walks. In: NIPS, pp. 873–879 (2000)
9. Metzler, D., Dumais, S., Meek, C.: Similarity Measures for Short Segments of Text. In: Proceedings of ECIR, pp. 16–27 (2007)
10. Mooter, http://www.mooter.com
11. Newman, M., Girvan, M.: Finding and Evaluating Community Structure in Networks. Phys. Rev. E 69(6), 026113 (2004)
12. Ni, X., Xue, G., Ling, X., Yu, Y., Yang, Q.: Exploring in the Weblog Space by Detecting Informative and Affective Articles. In: Proceedings of WWW, pp. 281–290 (2007)
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford University (1998)
14. Public Opinion, http://en.wikipedia.org/wiki/Public_opinion
15. Public Opinion Channel, http://yq.people.com.cn/CaseLib.htm
16. Technorati, http://technorati.com
17. Vivisimo, http://vivisimo.com
18. Wan, X., Yang, J., Xiao, J.: Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction. In: Proceedings of ACL, pp. 552–559 (2007)
19. WordNet, http://wordnet.princeton.edu
20. Yih, W., Meek, C.: Improving Similarity Measures for Short Segments of Text. In: Proceedings of AAAI, pp. 1489–1494 (2007)
21. Zeng, H., He, Q., Chen, Z., Ma, W., Ma, J.: Learning to Cluster Web Search Results. In: Proceedings of SIGIR, pp. 210–217 (2004)
22. Zhang, M., Ye, X.: A Generation Model to Unify Topic Relevance and Lexicon-based Sentiment for Opinion Retrieval. In: Proceedings of SIGIR, pp. 411–418 (2008)
23. Zhang, W., Yu, C., Meng, W.: Opinion Retrieval from Blogs. In: Proceedings of CIKM, pp. 831–840 (2007)