Article

JIS

Journal of Information Science
2017, Vol. 43(1) 54–74

DOI: 10.1177/0165551515617374
journals.sagepub.com/home/jis
SAGE

Exploring performance of clustering methods on document sentiment analysis

Baojun Ma
School of Economics and Management, Beijing University of Posts and Telecommunications, China

Hua Yuan
School of Management and Economics, University of Electronic Science and Technology of China, China

Ye Wu
School of Science, Beijing University of Posts and Telecommunications, China

Abstract
Clustering is a powerful unsupervised tool for sentiment analysis from text. However, the clustering results may be affected by any step of the clustering process, such as data pre-processing strategy, term weighting method in Vector Space Model and clustering algorithm. This paper presents the results of an experimental study of some common clustering techniques with respect to the task of sentiment analysis. Different from previous studies, in particular, we investigate the combination effects of these factors with a series of comprehensive experimental studies. The experimental results indicate that, first, the *K*-means-type clustering algorithms show clear advantages on balanced review datasets, while performing rather poorly on unbalanced datasets by considering clustering accuracy. Second, the comparatively newly designed weighting models are better than the traditional weighting models for sentiment clustering on both balanced and unbalanced datasets. Furthermore, adjective and adverb words extraction strategy can offer obvious improvements on clustering performance, while strategies of adopting stemming and stopword removal will bring negative influences on sentiment clustering. The experimental results would be valuable for both the study and usage of clustering methods in online review sentiment analysis.

Keywords
Clustering; data pre-processing; sentiment analysis; term weighting model

1. Introduction

With the prevalence of Internet and Web 2.0 technology, Internet hosts have accumulated a huge amount of data of text, pictures, audio, video and so on. Currently, the majority of this data is in the form of text. Thus, *sentiment analysis* (also known as *opinion mining*), which commonly refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials [1], has become a valuable area of research and has attracted many researchers from both academia and industry.

However, analysing the latent opinions and sentiments from massive documents is still a challenging task. In the literature, sentiment analysis is treated as a machine learning process [2, 3] that aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. Generally, three main phases of data pre-processing, vector space modelling (VSM) [4] and sentiment analysing (learning) are involved.

Corresponding author:
Hua Yuan, School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 611731, China.
Email: yuanhua@uestc.edu.cn

In data pre-processing, word segmentation is used to tokenize the document text into words at the initial stage of natural language processing task [5], which divides a string of written language into some component units. Sequentially, data cleaning is necessary for data pre-processing because not all the components are useful for the sentiment analysis task. Normally, the noise phrases, stopwords and meaningless symbols are removed [6].

After data pre-processing, VSM is a common approach for representing text documents as vectors. In VSM modelling, a document is conceptually represented by a vector of terms extracted from the document, with associated weights indicating the importance of the terms in the document and within the whole document collection [7]. If we have a large collection of N documents, and hence a large number of document vectors as $d_j = \{w_{1j}, w_{2j}, \ldots, w_{Nj}\}$, each dimension of d_j corresponds to the measurement of the weight of a separate term and can be determined in many ways, for example, the so-called TF-IDF method. VSM performs well on tasks that involve measuring the similarities of meanings between words, phrases and documents. Based on this, the classic supervised/semi-supervised and unsupervised learning methods can be conducted to automate sentiment analysis.

Supervised learning generates a model that can map inputs to desired outputs (also called labels), which are labelled by human experts according to some previously selected training samples. The most common supervised learning model is usually formulated as a two-class sentiment classification problem, that is, *positive* and *negative* [2]. Since it is a textual classification problem, any supervised learning method can be applied, for example, Naïve Bayes classification, and Support Vector Machines. In contrast, in an unsupervised learning model, the labels are not known during training. As a typical unsupervised learning method, clustering, which tries to find the natural clusters in the data by calculating the distances or similarities from the centres of the clusters, is especially useful for organizing documents to improve information retrieval. For example, clustering methods can be used to automatically group the retrieved documents into a list of meaningful categories [8]. The main advantage of clustering over classification is that it is adaptable to changes and helps single out useful features that distinguish different groups [9].

While conducting clustering analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups. However, similar to other intelligent methods, the analysis results generated by a clustering method would be probably affected by some intermediate steps, such as pre-processing strategy, term weighting model and clustering algorithm. It is valuable for the sentiment analysis tasks to make it clear:

- What kind of clustering algorithms are more effective for clustering-based sentiment analysis and what are their performance differences between datasets?
- What types of term weighting models are more suitable for review representation in the light of sentiment clustering?
- Does each of the data pre-processing steps be necessary for review clustering?

In this work, we conduct a set of comprehensive comparison studies of the online review sentiment clustering problem from a combined perspective of data pre-processing, VSM modelling and clustering algorithm and intend to examine the three questions with respect to online review clustering. The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 sketches out the research framework and approaches in detail. Section 4 shows the experimental results and Section 5 elaborates the concerns of discussion. This paper is concluded in Section 6.

2. Related work

2.1. Sentiment analysis

Generally, sentiment analysis aims to determine the opinions, sentiments, evaluations, attitudes, emotions and all the characteristics of a speaker or a writer with respect to some topics (i.e. entities like products, organizations, individuals, events) or the overall contextual polarity of a document.

There was no special interest in this field before the year 2000; however, since the spread of commercial applications with huge amounts of user-generated contents in online social media, the perspective on analysing opinions has changed radically [10]. In particular, the rise of social media such as blogs and online reviews has fuelled interest in sentiment analysis. With the proliferation of reviews, ratings, recommendations and other forms of online expression, online opinion has turned into a kind of virtual currency for businesses looking to market their products, identify new opportunities and manage their reputations [10].

Based on the existing research, sentiment analysis has been investigated mainly at three levels depending on the level of precision or interest required:

- *Document analysis* – determining if an entire document expresses a positive or negative opinion.
- *Sentence analysis* – determining if a sentence is positive, negative or neutral.
- *Entity and aspect or feature analysis* – determining if that is an opinion, on what is about and its polarity.

2.2. Sentiment clustering

Existing approaches to sentiment analysis can be grouped into four main categories: keyword spotting, lexical affinity, statistical methods and concept-level techniques [11].

- *Keyword spotting* classifies texts by affect categories based on the presence of unambiguous affect words such as happy, sad, afraid and bored [12].
- *Lexical affinity* not only detects obvious affect words, it also assigns arbitrary words a probable 'affinity' to particular emotions [13].
- *Statistical methods* leverage on elements from machine learning.
- *Concept-level approaches* leverage on elements from knowledge representation [14].

To analysis the sentiment automatically and more reasonably, unsupervised machine learning methods have drawn wide attention, for example, the clustering methods. In general, there are two kinds of common algorithms:

- *The hierarchical based algorithms*, which include single link, complete linkage, group average and Ward's methods. The hierarchical-based algorithm is suitable for browsing. However, such an algorithm usually suffers from efficiency problems.
- *The centroid based algorithms*, which is developed using the *K*-means algorithm and its variants.

In Su and Markert [12], a clustering method was proposed to map implicit aspect expressions, which were assumed to be sentiment words, to their corresponding explicit aspects. Some algorithms have also been proposed to use domain knowledge or constraints to guide topic modelling to produce better topic clusters [13]. Zhai et al. [14] presented a method to cluster the product features. The latest progress in sentiment clustering concerns dynamic sentiment analysis [15].

Since there are so many clustering methods for data processing in literature and they share the common characteristics and advantages of clustering, studying the feasibility of each method for different clustering tasks is an interesting work for text mining. Following the preliminary work presented by Steinbach et al. [16], some research has been done on comparing the common techniques used in document clustering [2, 3]. However, for the clustering method used in sentiment analysis, it has not been well understood, and will be investigated in this paper in detail.

3. Experimental setup

In this section, we describe our experimental setup for investigating the influences of related factors for online reviews clustering. The whole experimental processing, mainly including data pre-processing, VSM modelling and clustering, is illustrated in Figure 1. To that end, we selected 18 clustering algorithms and eight online review benchmark datasets; these are described in Sections 3.4 and 3.1 in detail, respectively. For each dataset we considered three steps of data pre-processing: adjective and adverb words extraction, words stemming and stopword removal, which resulted in eight kinds of data inputs for VSM modelling. For each kind of data input, we applied six types of term weighting models to construct the review vectors and cosine similarity was used to represent the similarity between review vectors. This gave a total of 6912 ($8 \times 8 \times 6 \times 18$) clusterings, which were evaluated using the accuracy measure. All experiments were conducted on a computer with a 3.10 GHz Intel Core 2 processor, 8 GB RAM, and 64-bit Windows 7. All algorithms in data pre-processing as well as VSM modelling were implemented in JAVA and clustering algorithms were implemented in MATLAB and CLUTO toolkit [17]. The following subsections detail the specific datasets, data pre-processing and clustering algorithms we used.

3.1. Datasets

We used a total of eight benchmarked datasets in our experiments, which have been utilized in numerous publications [7, 18–40] and may thus be considered as standard test sets for sentiment analysis. Table 1 summaries their characteristics.

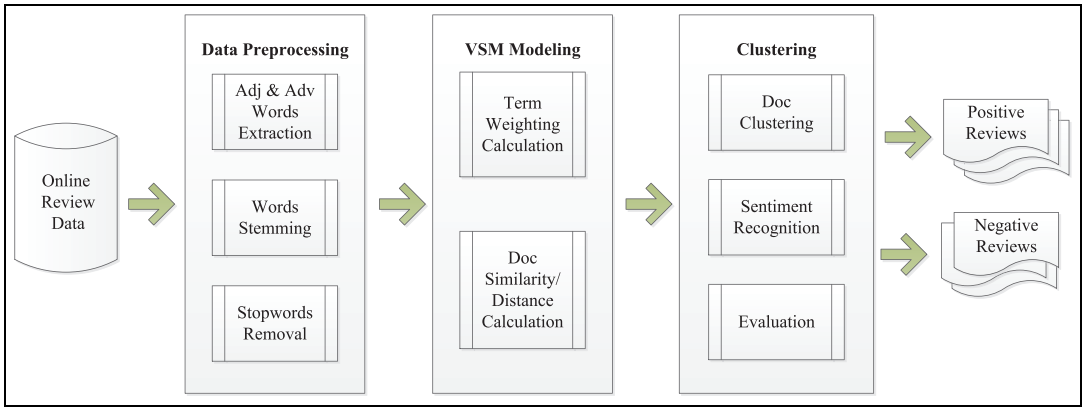

Figure 1. Experimental processing for online review clustering.

Table 1. The benchmarked datasets used in our experiments.

ID	Name	Reference	Download URL
D1	Polarity Dataset V2.0	Huang and Croft [7], Argamon et al. [19], Boiy and Moens [24], Li and Liu [32], Pang and Lee [37], Sindhwani and Melville [39], Venkatasubramanian et al. [40]	http://www.cs.cornell.edu/People/pabo/movie-review-data
D2	Sentence Polarity Dataset v1.0	Agarwal et al. [18], Goldberg and Zhu [28], Pang and Lee [38]	http://www.cs.cornell.edu/People/pabo/movie-review-data
D3	Amazon product reviews for Books	Blitzer et al. [22, 23], Dredze et al. [26], Gao and Li [27], Li et al. [33], Mansour et al. [34]	http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
D4	Amazon product reviews for DVDs		
D5	Amazon product reviews for Electronics		
D6	Amazon product reviews for Kitchen		
D7	TripAdvisor-15763	Baccianella et al. [20, 21]	http://patty.isti.cnr.it/~baccianella/reviewdata/corpus/TripAdvisor_corpus.zip
D8	Amazon-83713	Baccianella et al. [21], Jindal and Liu [29, 30], Jindal et al. [31], Mukherjee et al. [35]	http://patty.isti.cnr.it/~baccianella/reviewdata/corpus/Amazon_corpus.zip

The review collection Polarity Dataset V2.0 was constructed by Pang and Lee [37] and consists of 1000 positive and 1000 negative reviews, which were taken from the IMDB movie review archives.[1] The Sentence Polarity Dataset v1.0 collection consists of 5331 positive and 5331 negative movie-review 'snippets' (a striking extract usually one sentence long) downloaded from www.rottentomatoes.com. The collections Amazon product reviews for Books, DVDs, Electronics and Kitchen (i.e. datasets 3–6) are from the Multi-Domain Sentiment Dataset found in Blitzer et al. [23]. These four datasets consists of review texts and rating labels, taken from amazon.com product reviews within four different categories (i.e. book, DVD, electronics and kitchen), where each domain contains 1000 positive and 1000 negative reviews. The TripAdvisor-15763 collections were constructed by Baccianella et al. [20] crawled from TripAdvisor,[2] which is one of the most popular online review sites for tourism-related activities. It contains 15,763 hotel reviews in total, including 12,387 positive reviews with rating larger than 3 and 1800 negative ones with rating smaller than 3. The final review collections, called Amazon-83713, is composed of 83,713 reviews from the Amazon Web set, which is the small subset of the Amazon dataset (consisting of more than 5 million reviews) originally built by Jindal and Liu [29] for spam review detection purposes, and contains all the reviews in the sections of MP3, USB, GPS, Wireless 802.11, Digital Camera and Mobile Phone. For review collections Sentence Polarity Dataset v1.0, TripAdvisor-15763 and

Table 2. Different types of data pre-processing.

Pre-processing substep	111	110	101	100	011	010	001	000
Adjective and adverb word extraction	√	√	√	√	×	×	×	×
Word stemming	√	√	×	×	√	√	×	×
Stopword removal	√	×	√	×	√	×	√	×

Table 3. Preliminary notations for term weighting.

Notation	Description
D	Document set or collection
d_i	Document i in D
T	Complete term set of D
t_j	Term j in D
N	Number of documents in D, $N = \|D\|$
tf_{ij}	Term frequency of t_j in d_i
df_i	Document frequency of t_j, that is, the number of documents containing t_j
tf_i	Term frequency of t_j in D, $tf_j = \sum_i tf_{ij}$
dl_i	Document length of d_i, $dl_i = \sum_j tf_{ij}$
$avgdl$	Average document length, $avgdl = \sum_i dl_i / N$
$sigmatf$	Total number of term frequency, $sigmatf = \sum_j tf_j = \sum_i dl_i = \sum_j \sum_i tf_{ij}$

Amazon_83713, we randomly selected 1000 positive and 1000 negative reviews from each original dataset for clustering.

3.2. Data pre-processing

In the experiments, we took three sequential substeps into consideration, including adjective and adverb words extraction, words stemming as well as stopword removal. In order to determine whether each of the three substeps was having a positive effect on review clustering, we generated eight different types of data pre-processing, shown in Table 2, which would result in eight distinct data inputs for the VSM modelling stage on each dataset. For instance, the '101-type' data pre-processing means the adjective and adverb words extraction and stopword removal process would be conducted without words stemming.

To extract adjective and adverb words, a part-of-speech tagger developed by Stanford University was used to tag the reviews [41]. If this step was conducted, words that were not tagged as being either an adjective or an adverb were eliminated. Then, the step of words stemming was done by applying Porter's algorithm [42]. To remove stopwords, we utilized the stopword list built by Gerard Salton and Chris Buckley for the experimental SMART information retrieval system at Cornell University, which contains 571 stopwords [43–45].

3.3. Term weighting models

In the experiments, we treated the online reviews as web documents, which were usually regarded as 'bags of words'. Let $D = \{d_1, d_2, \ldots, d_N\}$ be a set of documents or reviews we want to cluster, and d_i be the ith document or review of D. Each d_i is composed of several keywords or terms and let $T = \{t_1, t_2, \ldots, t_M\}$ be the complete term set of D. Table 3 lists some useful preliminary notations.

Each d_i, it could be represented using the standard vector space model representation:

$$d_i = [w_{i1}, w_{i2}, \ldots, w_{iM}] \quad (1)$$

where w_{ij} is the weight of term t_j to document d_i. The document clustering algorithms are based on the weighting vectors or the similarity matrices generated. Table 4 illustrates six term weighting models selected to be utilized in our experiments, which are denoted as Binary, TF, TF_IDF, BM25, DPH_DFR and H_LM for short in the following discussion.

Table 4. Term weighting models used in our experiments.

Term weighting model	Equation	Reference
Presence (Binary)	$w_{ij} = \begin{cases} 1 & \text{if } tf_{ij} > 0; \\ 0 & \text{otherwise.} \end{cases}$	Baeza-Yates and Ribeiro-Neto [46]
Term frequency (TF)	$w_{ij} = tf_{ij}$.	Salton et al. [45, 47]
TF_IDF	$w_{ij} = tf_{ij} \times \log \frac{N}{df_j}$.	Jones [48], Robertson and Jones [49]
Okapi's BM25 (BM25)	$w_{ij} = \frac{tf_{ij}(k_1+1)}{tf_{ij} + k_1\left((1-b) + b \cdot \frac{dl_i}{avgdl}\right)} \times \log \frac{N}{df_j}$.	Robertson et al. [50]
DPH divergence from randomness (DPH_DFR)	$w_{ij} = \begin{cases} 0 & \text{if } tf_{ij} = 0; \\ \frac{\left(1 - \frac{tf_{ij}}{dl_i}\right)^2}{tf_{ij}+1} \cdot \{tf_{ij} \cdot \log\left(\frac{tf_{ij} \cdot avgdl}{dl_i} \cdot \frac{N}{tf_j}\right) + 0.5 \cdot \log[2\pi \cdot tf_{ij} \cdot (1 - \frac{tf_{ij}}{dl_i})]\} & \text{otherwise.} \end{cases}$	Amati et al. [51]
Hiemstra language model (H_LM)	$w_{ij} = \log\left(1 + \frac{\lambda \cdot tf_{ij} \cdot sigmatf}{(1-\lambda) \cdot df_j \cdot dl_i}\right)$.	Hiemstra [52]

In the BM25 weighting model, b and k_1 are parameters that are tuned, with $k_1 \geq 0$ and $0 \leq b \leq 1$ and in H_LM model, λ is also the parameter with $0 \leq \lambda \leq 1$. For our experiments, we used the default values for the above three parameters like the previous literature, that is, $k_1 = 1.2$, $b = 0.75$ and $\lambda = 0.15$ [50, 52, 53].

3.4. Clustering algorithms

For the 18 clustering algorithms used in our experiments, all but the methods based on Zhao and Karypis [54–56] were independently implemented by one of the authors using the MATLAB environment. Where possible, the implementations were validated through comparisons with previously published results. The algorithms utilizing the objective functions from Zhao and Karypis [54–56] were performed using the authors' own clustering toolkit.[3]

The reason why we selected these clustering algorithms was based on the following criteria: (1) were they well-established algorithms; (2) did they take a pre-specified number of clusters as a parameter and produce hard clusters; (3) did the selected set of algorithms cover a breadth of the well-established methods for document clustering; and (4) together, did the set of algorithms include those methods reported to produce good results in previous research? Table 5 lists the 18 clustering algorithms we selected. Below we would give a brief description of each method.

Our *Kmeans* algorithm is based on Lloyd's method [57] with the initial centroids being selected randomly from the vectors of each dataset. We used the direct k-way clustering method with I2 criterion function (i.e. *Direct-I2* for short) in the CLUTO toolkit to implement the *Kmeans* algorithm, in which the basic *Kmeans* algorithm was run 10 times and only the best result was kept according to the *Kmeans* internal objective function (i.e. I2 criterion function). *RB-Kmeans* repeated splits the dataset using *Kmeans* with global optimization finally, which was implemented using the RBR method with I2 criterion function (i.e. *RBR-I2* for short) in the CLUTO toolkit. In *RB-Kmeans*, binary splitting is used, with the largest remaining cluster split at each iteration [16]. The *PAM* algorithm using medoid swapping [58] was chosen in our study as a different approach to optimizing the same objective function as our *Kmeans* algorithm. The *CLARANS* algorithm is a typical sampling-based k-medoids partitioning clustering method scaling well for large datasets, which combines the sampling technique with *PAM* and draws a sample with some randomness in each step of the search [59].

Three varieties of spectral clustering algorithms have been selected in our experiments. For each of these three methods the weighted adjacency matrix W was generated through an r-nearest neighbour scheme with cosine similarity using $r = 20$. *Spect-Un* clusters on the k eigenvectors of the Laplacian $L = D - W$, where D is the degree matrix of W [60]. For details on the Laplacian for *Spect-RW* see Shi and Malik [61], and for *Spect-Sy* see Ng et al. [62]. The clustering of the eigenvectors for all three methods was done using the *Kmeans* algorithm.

PCA is a common dimensionality-reduction technique based on singular value decomposition [63]. For our *PCA-Kmeans* algorithm, we first applied *PCA* to produce a reduced document feature space. Dimensionality reduction was followed by the application of the *Kmeans* algorithm. Non-negative matrix factorization is a relative recent document

Table 5. The clustering algorithms used in our experiments.

ID	Clustering algorithm	Short name	Reference
1	*K*-means	*Kmeans* (*Direct-I2*)	Lloyd [57]
2	Repeated bisecting *K*-means	*RB-Kmeans* (*RBR-I2*)	Steinbach et al. [16]
3	Partition around medoids	*PAM*	Kaufman and Rousseeuw [58]
4	Clustering large applications based upon randomized search	*CLARANS*	Ng and Han [59]
5	Unnormalized spectral	*Spect-Un*	Luxburg [60]
6	Random walk spectral	*Spect-RW*	Shi and Malik [61]
7	Symmetric spectral	*Spect-Sy*	Ng et al. [62]
8	Principle component analysis + *K*-means	*PCA-Kmeans*	Pearson [63]
9	Non-negative matrix factorization	*NC-NMF*	Xu et al. [5]
10	Unweighted pair group method	*UPGMA* (*Agglo-upgma*)	Kaufman and Rousseeuw [58]
11	Single linkage	*Slink* (*Agglo-Slink*)	Jain et al. [6]
12	Complete linkage	*Clink* (*Agglo-Clink*)	Jain et al. [6]
13	Repeated bisecting H1 with global optimization	*RBR-H1*	Zhao and Karypis [54–56]
14	Direct H1	*Direct-H1*	Zhao and Karypis [54–56]
15	Agglomerative I2	*Agglo-I2*	Zhao and Karypis [54–56]
16	Agglomerative H1	*Agglo-H1*	Zhao and Karypis [54–56]
17	Agglomerative cluster-weighted single-link	*Agglo-WSlink*	Zhao and Karypis [54–56]
18	Agglomerative cluster-weighted complete-link	*Agglo-WClink*	Zhao and Karypis [54–56]

clustering method that has been shown to be highly effective and the variety we implemented is discussed by Xu et al. [5]. Note that we used *NC-NMF* as the authors showed it to be more effective than simple *NMF*.

UPGMA [58], *Slink* [6] and *Clink* [6] are from the same family of agglomerative clustering algorithms. In each of these methods every document begins as a singleton cluster and clusters are progressively merged with the best similarity. In *Slink* similarity between clusters is equal to the similarity of their closest documents. In *Clink* the farthest documents are used. In *UPGMA* the average similarity between all documents is used. Cosine similarity was used for all three of these methods. We respectively utilized *Agglo-upgma, Agglo-Slink* and *Agglo-Clink* in CLUTO toolkit to implement the above three agglomerative clustering algorithms.

Finally, we selected six other clustering methods from Zhao and Karypis [54–56] besides those introduced above implemented in CLUTO toolkit (e.g. *Direct-I2, RBR-I2, Agglo-upgma, Agglo-Slink*), which we refer to as *RBR-H1, Direct-H1, Agglo-I2, Agglo-H1, Agglo-WSlink* and *Agglo-WClink*. We selected three distinct optimization methods – repeated bisection (RBR), direct partitional (Direct) and agglomerative (Agglo) – as well as four different objective functions, H1, I2, WSlink and WClink. The I2 function is essentially the *Kmeans* objective function except that any similarity metric may be used in the calculation. The H1 function is I1/E1, where E1 is an objective function based around minimizing the weighted similarity of cluster centroids from the centroid of the whole dataset. The WSlink function is the cluster-weighted single-link criterion function and the WClink function serves as the cluster-weighted complete-link criterion function. For details on their exact implementations, readers can consult Zhao and Karypis [54–56] as we used the authors' own clustering toolkit to perform these algorithms.

The above algorithms are by no means a full list of document clustering methods. In addition to just the variants of what we used, there are some totally different methods, such as those based on frequent itemsets [64, 65], the information bottleneck [66], numerous model-based approaches [8, 67–70], and so on. Still, we believe that we implemented a sufficient set of algorithms to perform our experiments.

3.5. Evaluation measure

After all the reviews in one dataset were divided into two clusters – cluster 1 and cluster 2 – we needed to evaluate the clustering accuracy. However, we did not know which cluster was positive and which was negative. Fortunately, we knew the actual class of each review based on the label or rating, and therefore a confusion matrix could be constructed as shown in Table 6, in which *a, b, c* and *d* are the numbers of reviews in each cell. Thus, if $(a + d) > (b + c)$, cluster 1 will be regarded as the positive cluster, otherwise cluster 2 will be regarded as the positive cluster. Consequently, the accuracy measure for review clustering can be calculated as

Table 6. Confusion matrix of review clustering.

	Cluster 1	Cluster 2
Actual positive	*A*	*B*
Actual negative	*C*	*D*

$$\text{accuracy} = \begin{cases} \dfrac{a+d}{a+b+c+d} & \text{if } (a+d) \geq (b+c) \\ \dfrac{b+c}{a+b+c+d} & \text{if } (a+d) < (b+c) \end{cases} \quad (2)$$

Certainly, in real applications, the actual class of online reviews is unknown. Li and Liu [32] proposed to choose solid polarity reviews to generate a positive seed set and a negative seed set to solve this problem, in which the authors proved that the possibility of incorrect major direction is extremely low when the size of seed set is sufficiently large. For details on the discussion about the issue of sentiment recognition, readers can consult Li and Liu [32].

4. Experimental results

Rather than considering the influences of clustering algorithms, term weighting and data pre-processing together at once, we chose to examine the following three questions we believe are crucial to online review clustering:

- Which kinds of clustering algorithms are more effective for clustering-based sentiment analysis?
- Which types of term weighting models are more suitable for review representation?
- Is each of the data pre-processing steps necessary for review clustering?

4.1. Results on clustering algorithms

To determine which kinds of clustering algorithms are more effective for clustering-based sentiment analysis, we first took the accuracy measures on the 6912 ($8 \times 6 \times 18 \times 8$) tuples of (pre-processing, term weighting, algorithm, dataset) and collapsed them by averaging each accuracy measures over data pre-processing strategies and term weighting models. Furthermore, we also considered the highest accuracy measures for each clustering algorithm on each dataset. This gave 48 tuples of (algorithm, dataset) for each of the previous two situations, which are illustrated in Figure 2. In Figure 2, the lines of *Kmeans* and *RB-Kmeans* coincide with each other on the whole, as do those of *RBR-H1* and *Direct-H1*.

To see the differences between clustering performances more clearly, from these 48 tuples we derived three tables comparing the 18 clustering algorithms: (1) by dataset and average over all clustering algorithms for that dataset (Table 7); (2) by dataset and the best performance of each clustering algorithm for that dataset (Table 8); and (3) by clustering main type (Table 9). The last columns of Tables 7 and 8 display the average rank (i.e. AvgRank) for each algorithm. The rank score is calculated by ranking all techniques according to their performance on each dataset, rank 1 indicating the best performance and rank 18 the worst. The AvgRanks are then obtained by averaging the ranks across all eight datasets.

Figure 2 and the AvgRank column in Table 7 indicate that, on average, four of the 18 algorithms (i.e. *Kmeans, RB-Kmeans, RBR-H1* and *Direct-H1*) show clear advantages over the other 14 methods on clustering accuracy. However, the *CLARANS* algorithm is actually somewhat better than the above four best methods and substantially better than other 13 algorithms on D2 and D3. Table 8 shows the best performance for each dataset and clustering algorithm. One can see that the results in Table 8 are mostly consistent with those in Table 7.

The average and best performances by algorithm presented in Table 9 are split into partitional and hierarchical groups (note that *RB-Kmeans* and other repeated bisection methods are placed in the hierarchical section as they generate hierarchies of clusters, even though the splitting decision at each level is based on partitioning). It is clear from Table 9 that the clustering performances are not divisible along partitional vs hierarchical lines. For instance, *Kmeans, Direct-H1, RB-Kmeans* and *RBR-H1* gain the highest performances, the former two being partitional and the latter two being hierarchical.

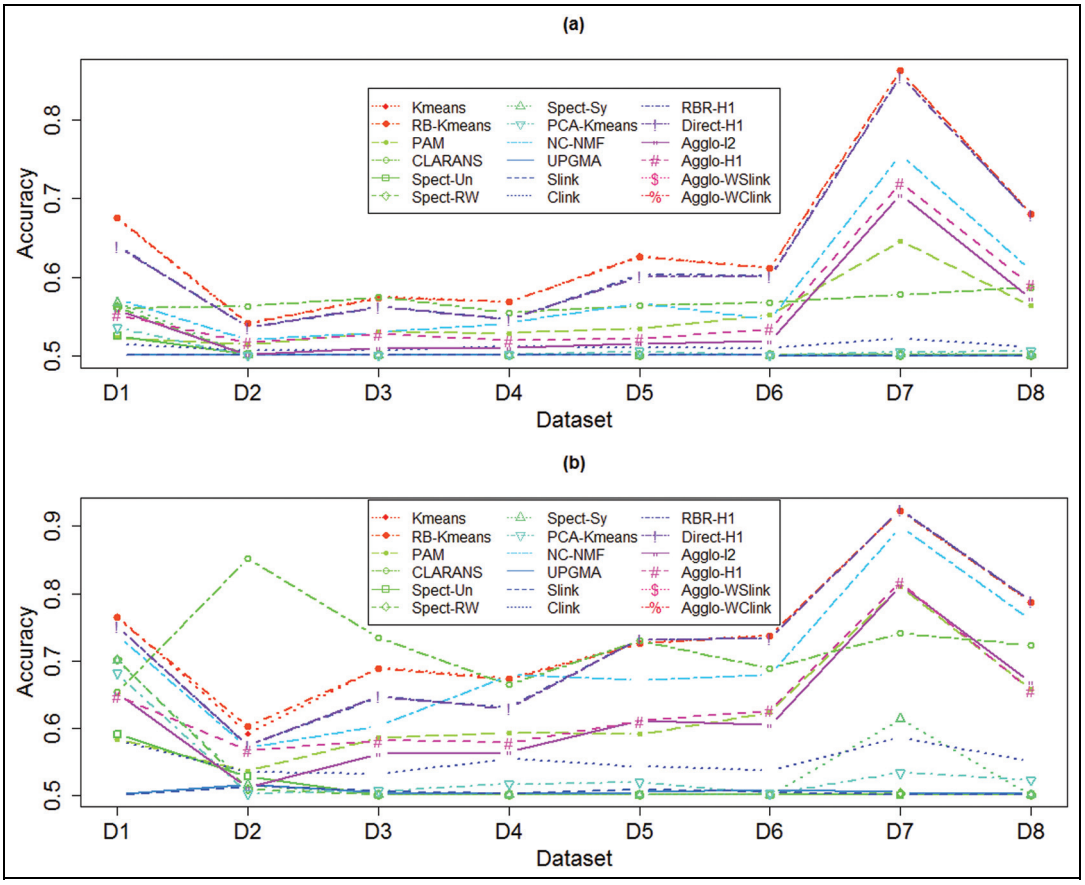

Figure 2. The average (a) and highest (b) performance of each clustering algorithm on each dataset. Figure is reproduced in colour in online version.

Table 7. The percentage difference of average accuracy for each clustering algorithm over the best performance by dataset.

Algorithm	D1	D2	D3	D4	D5	D6	D7	D8	AvgRank
Kmeans	0	−3.8	−0.1	0	0	0	0	0	1.6
RB-Kmeans	0	−3.7	−0.1	0	−0.1	0	0	0	2.3
PAM	−22.5	−8.4	−7.7	−7.1	−14.6	−9.7	−11.8	−25.1	8.0
CLARANS	−17.1	0	0	−2.4	−10	−7.3	−11.4	−33	5.0
Spect-Un	−22.1	−10.8	−13	−12	−20.1	−18.2	−23.6	−42	14.8
Spect-RW	−16.8	−10.9	−13	−12	−20.1	−18.2	−23.6	−42	14.8
Spect-Sy	−15.9	−10.9	−13	−12	−20.1	−18.2	−23.4	−41.7	14.4
PCA-Kmeans	−20.6	−11	−12.9	−11.7	−19.2	−18.2	−22.9	−41.5	13.4
NC-NMF	−15.3	−7.3	−7.9	−4.7	−9.6	−10.7	−6.5	−12.3	5.8
UPGMA	−25.9	−10.9	−12.9	−11.9	−19.9	−18.1	−23.6	−42	14.3
Slink	−25.9	−10.8	−12.9	−12	−20	−18.2	−23.6	−42	14.1
Clink	−23.8	−9.9	−11.8	−10.1	−18.4	−16.7	−21.9	−39.5	10.3
RBR-H1	−5.3	−4.5	−2.1	−3.9	−3.8	−1.6	−0.7	−0.8	3.4
Direct-H1	−5.4	−4.5	−2.1	−3.9	−4.1	−1.9	−0.7	−0.8	3.9
Agglo-I2	−17.4	−10.8	−11.3	−10.1	−17.6	−15.3	−12.2	−18.2	9.3
Agglo-H1	−18.2	−8	−8.1	−8.5	−16.7	−12.5	−8.3	−16.5	7.6
Agglo-WSlink	−25.9	−10.8	−12.9	−12	−20	−18.2	−23.6	−42	14.1
Agglo-WClink	−23.8	−9.9	−11.8	−10.1	−18.4	−16.7	−21.9	−39.5	10.3

In order to investigate the relationship among the clustering performances of the selected 18 algorithms more precisely, we also conducted statistical significance tests on paired comparison on clustering algorithms' values of accuracy over all eight benchmarked datasets (see Table 10). For each two compared clustering algorithms, there are in total 384

Table 8. The percentage difference of the highest accuracy for each clustering algorithm over the best performance by dataset.

Algorithm	D1	D2	D3	D4	D5	D6	D7	D8	AvgRank
Kmeans	0	− 30.6	− 8.6	− 0.4	− 3.8	0	− 0.2	− 0.9	2.3
RB-Kmeans	− 0.3	− 29.3	− 8.9	− 1.5	− 3.8	0	− 0.2	− 0.9	2.5
PAM	− 24	− 37.1	− 20.3	− 10.8	− 19.1	− 15.5	− 12.4	− 16.7	8.3
CLARANS	− 14.7	0	0	0	0	− 6.6	− 19.9	− 8.4	4.1
Spect-Un	− 23	− 38	− 31.8	− 24.7	− 31.4	− 32.1	− 45.9	− 36.6	14.9
Spect-RW	− 8.4	− 40.3	− 31.8	− 24.7	− 31.4	− 32.1	− 45.7	− 36.6	14.5
Spect-Sy	− 8.6	− 39.6	− 31.8	− 24.7	− 31.4	− 32.1	− 33.7	− 36.6	13.8
PCA-Kmeans	− 11.2	− 41	− 31	− 22.3	− 29.3	− 31.9	− 42.4	− 33.8	12.8
NC-NMF	− 4.5	− 33	− 18.7	− 3.3	− 8	− 7.8	− 2.8	− 3.6	5.4
UPGMA	− 34.5	− 39.4	− 31.4	− 24.4	− 30.8	− 31.1	− 45.5	− 36.4	13.8
Slink	− 34.7	− 40.4	− 31.1	− 24.5	− 30.6	− 31.5	− 45.9	− 36.5	14.5
Clink	− 24.3	− 37.3	− 27.6	− 16.6	− 25.6	− 27.3	− 36.7	− 30.3	10.5
RBR-H1	− 1.9	− 32.9	− 12	− 5.1	− 3.8	− 0.5	0	0	3.0
Direct-H1	− 1.9	− 32.9	− 11.9	− 5.3	− 4.1	− 0.5	0	0	3.4
Agglo-I2	− 15	− 40	− 23.5	− 15.1	− 16.2	− 18	− 12.2	− 17.2	9.3
Agglo-H1	− 15.6	− 33.5	− 21.9	− 12.9	− 20.9	− 15.1	− 11.7	− 17.1	8.0
Agglo-WSlink	− 34.7	− 40.4	− 31.1	− 24.5	− 30.6	− 31.5	− 45.9	− 36.5	14.5
Agglo-WClink	− 24.3	− 37.3	− 27.6	− 16.6	− 25.6	− 27.3	− 36.7	− 30.3	10.5

Table 9. Average and best performances for clustering algorithms split into two groups.

Main type	Algorithm	Average accuracy	Highest accuracy
Partitional	*Kmeans*	64.26%	92.35%
	Direct-H1	62.78%	92.5%
	NC-NMF	58.04%	89.95%
	CLARANS	56.88%	85.2%
	PAM	54.96%	81%
	Spect-Sy	50.94%	70.05%
	Spect-RW	50.86%	70.2%
	PCA-Kmeans	50.72%	68.1%
	Spect-Un	50.39%	59.05%
Hierarchical	*RB-Kmeans*	64.26%	92.35%
	RBR-H1	62.83%	92.5%
	Agglo-H1	56.09%	81.65%
	Agglo-I2	54.93%	81.25%
	Clink	51.20%	58.55%
	Agglo-WSlink	51.20%	58.55%
	UPGMA	50.10%	51.6%
	Slink	50.10%	50.8%
	Agglo-WClink	50.09%	50.8%

($6 \times 8 \times 8$) paired results of clustering accuracy to make comparisons. In Table 10, the symbol in each cell (e.g. row *a*, column *b*) represents the paired comparative result of statistical significance test for the clustering algorithm in row *a* with respect to that in column *b*. For instance, the symbol '$<<$' (highlighted in bold) in row 6 and column 7 in Table 10 implies that Algorithm 6 (i.e. *Spect-RW*) performed significantly worse than Algorithm 7 (i.e. *Spect-Sy*) with $p < 0.01$. Based on the results in Table 10, the 18 clustering algorithms can be divided into four groups according to their performances (see Table 11). The algorithm with smaller group id performs significantly better than that with larger group id. Moreover, the algorithms in the same group either perform no significant differences or possess relatively small performance differences on accuracy. From Tables 10 and 11, it is still easy to find that *Kmeans, RB-Kmeans, RBR-H1* and *Direct-H1* show clear advantages over the other 14 methods on clustering accuracy, which is consistent with the results in Tables 7 and 8, while based on Tables 10 and 11, *NC-NMF* performs significantly better than *CLARANS* with $p < 0.05$, which is inconsistent with that in Tables 7 and 8.

Table 10. Paired comparative results of statistical significance tests for each clustering algorithm on clustering accuracy.

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	—	≈	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >
2		—	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >	> > >
3			—	< < <	> > >	> > >	> > >	> > >	< < <	> > >	> > >	> > >	< < <	< < <	≈	< < <	> > >	> > >
4				—	> > >	> > >	> > >	> > >	<	> > >	> > >	> > >	< < <	< < <	> > >	≈	> > >	> > >
5					—	< < <	< < <	< < <	< < <	> > >	> > >	< < <	< < <	< < <	< < <	< < <	> > >	< < <
6						—	< <	≈	< < <	> > >	> > >	≈	< < <	< < <	< < <	< < <	> > >	≈
7							—	≈	< < <	> > >	> > >	≈	< < <	< < <	< < <	< < <	> > >	≈
8								—	< < <	> > >	> > >	< <	< < <	< < <	< < <	< < <	> > >	< <
9									—	> > >	> > >	> > >	< < <	< < <	> > >	> > >	> > >	> > >
10										—	≈	< < <	< < <	< < <	< < <	< < <	≈	< < <
11											—	< < <	< < <	< < <	< < <	< < <	≈	< < <
12												—	< < <	< < <	< < <	< < <	> > >	≈
13													—	> >	> > >	> > >	> > >	> > >
14														—	> > >	> > >	> > >	> > >
15															—	< < <	> > >	> > >
16																—	> > >	> > >
17																	—	< < <
18																		—

≈, No significant differences; > (<), significantly greater (less) than ($p < 0.05$); > > (< <), significantly greater (less) than ($p < 0.01$); > > > (< < <), significantly greater (less) than ($p < 0.001$).

Table 11. Partitional groups for selected clustering algorithms based on performances.

Group ID	Algorithm ID	Algorithm Short Name
Group 1	1	*Kmeans*
	2	*RB-Kmeans*
	13	*RBR-H1*
	14	*Direct-H1*
Group 2	9	*NC-NMF*
	4	*CLARANS*
	16	*Agglo-H1*
	15	*Agglo-I2*
	3	*PAM*
Group 3	7	*Spect-Sy*
	12	*Clink*
	18	*Agglo-WClink*
	6	*Spect-RW*
	8	*PCA-Kmeans*
	5	*Spect-Un*
Group 4	10	*UPGMA*
	11	*Slink*
	17	*Agglo-WSlink*

Table 12. The percentage difference of average accuracy for each weighting model over the best performance by dataset.

Weighting model	D1	D2	D3	D4	D5	D6	D7	D8	AvgRank
Binary	−3.1	−1.7	−0.4	−1.7	−1.8	−3.7	−5.1	−2.3	4.5
TF	−5.2	−1.4	−2.1	−1.5	−5.1	−4.6	−6.1	−4.1	5.5
TF_IDF	−8.2	−0.6	−0.2	−0.8	−3	−2.1	0	−2.7	4.0
BM25	−1	−0.4	0	0	−2.9	−0.5	0	0	1.9
DPH_DFR	0	0	−0.2	−0.6	0	−0.2	−2	−0.8	1.9
H_LM	−5.9	−0.6	−0.8	−1	−1.8	0	−1.7	−1.1	3.3

Table 13. The percentage difference of the highest accuracy for each weighting model over the best performance by dataset.

Weighting model	D1	D2	D3	D4	D5	D6	D7	D8	AvgRank
Binary	−5.8	−16.5	−2.2	−11.9	−9.8	−6.6	−7.9	−5.3	4.8
TF	−12.5	−26.5	−9.1	−3.6	−0.3	−9.9	−5.7	−13	4.9
TF_IDF	−4.4	−25	0	−2.1	−5.6	−10.4	−1	−4.6	3.8
BM25	0	−9.7	−6.1	0	−5.3	−5.1	−0.6	−2.7	2.3
DPH_DFR	−1.6	−2.8	−9.4	−6.2	−5.9	0	−1.9	−3.2	3.6
H_LM	−0.9	0	−7.8	−0.7	0	−1.5	0	0	1.8

4.2. Results on term weighting models

To determine which types of term weighting models are more suitable for review representation, we compared the clustering results on the six selected weighting models, which have been illustrated in three tables: (1) by dataset and average over all the weighting models for that dataset (Table 12); (2) by dataset and the best performance of each weighting model for that dataset (Table 13); and (3) by dataset and average over all the weighting models on four superior clustering algorithms (i.e. *Kmeans, RBR-Kmeans, RBR-H1* and *Direct-H1*) for that dataset (Table 14).

From the AvgRank columns of Tables 12–14, we can see that, on average, BM25, DPH_DFR and H_LM weighting models are somewhat better than TF_IDF, and notably better than Binary as well as TF. However, TF_IDF is actually sometimes somewhat better than H_LM and DPH_DFR, and performs similar results to BM25 on D2, D5 and D7.

Table 14. The percentage difference of average accuracy for each weighting model on four superior clustering algorithms over the best performance by dataset.

Weighting model	D1	D2	D3	D4	D5	D6	D7	D8	AvgRank
Binary	−7	−5.4	−4	−6.1	−7.8	−18	−12.2	−8.4	4.8
TF	−12	−5	−6.8	−6.6	−17	−17.8	−14.2	−12.8	5.6
TF_IDF	−15.9	0	−1.6	−1.9	−10.8	−7.7	−1.5	−3.5	3.5
BM25	−0.4	−1	0	0	−8.9	−4.6	−0.8	−2.1	2.5
DPH_DFR	0	−0.6	−1.1	−3.8	0	−3.8	−3.3	−0.6	2.4
H_LM	−8	−0.6	−2.7	−2.1	−2.7	0	0	0	2.3

Table 15. Paired comparative results of statistical significance tests for each weighting model on clustering accuracy.

Weighting model	Binary	TF	TF_IDF	BM25	DPH_DFR	H_LM
Binary	—					
TF	$<<<$	—				
TF_IDF	$\approx$	$>>>$	—			
BM25	$>>>$	$>>>$	$>>>$	—		
DPH_DFR	$>>>$	$>>>$	$>>>$	$\approx$	—	
H_LM	$>>$	$>>>$	$>>$	$<<<$	$<<<$	—

$\approx$, No significant differences; $>$ ($<$), significantly greater (less) than ($p < 0.05$); $>>$ ($<<$), significantly greater (less) than ($p < 0.01$); $>>>$ ($<<<$), significantly greater (less) than ($p < 0.001$).

At the same time, we also conducted statistical significance tests on paired comparison on weighting models' performances over all eight benchmarked datasets (see Table 15). For each two compared weighting models, there are in total 1152 ($18 \times 8 \times 8$) paired results of clustering accuracy to make comparisons. In Table 15, the symbol in each cell (e.g. row a, column b) represents the paired comparative result of statistical significance test for the weighting model in row a with respect to the weighting model in column b. For instance, the symbol of '$>>>$' (highlighted in bold) in row 4 and column 2 in Table 15 implies that BM25 model performed significantly better than the TF model with $p < 0.001$. From Table 15, it is easy to infer that the comparatively newly designed weighting models (e.g. BM25, DPH_DFR and H_LM) performed significantly better than the traditional ones (e.g. Binary, TF and TF_IDF). Moreover, TF_IDF and Binary models performed significantly better than TF. Although there are no significant differences between TF_IDF and Binary models, TF_IDF is actually somewhat better than Binary with the positive t-test value of 1.117.

A possible reason for BM25, DPH_DFR and H_LM weighting models' superiority to TF_IDF, Binary as well as TF is described as follows. The three comparatively newly designed weighting models were initially proposed in the domain of information retrieval to better show the degree of importance of the terms appearing in documents and thus to derive the relevance of a document to a given query more accurately by taking more elaborate information on terms, documents and document collection into consideration, rather than only term appearance in the traditional weighting models (e.g. Binary, TF and TF_IDF) [71]. For instance, BM25 incorporates the document length and the average length of all documents in the corpus as well as the term frequency normalization effect into its weighting model. DPH_DFR model regards the weight of a term in a document as the result of three components' interaction effect: the randomness model component, the aftereffect of sampling component and the document length normalization component, while H_LM model aims to solve the term weighting problem from the language modelling perspective by considering Jelinek–Mercer smoothing method, the document length and total number of term frequency in the entire document collection. Based on the above issues, several research works have shown the superiority of these comparatively newly designed weighting models to the traditional ones on the performance of information retrieval [50–52] and general document clustering [32] by conducting comprehensive experiments. Although sentiment clustering is not the same as general document clustering, owing to the fact that the comparatively newly designed weighting models possess better capabilities in representing documents precisely, it is no surprise that they perform better in our experiments.

Moreover, another additional evidence of the superiority of the three comparatively newly designed weighting models is apparent from a nearest neighbour analysis. For each dataset and weighting model we calculated the percentage of r-nearest neighbours, per online review, that share the same cluster label as that online review. Figure 3 presents the results

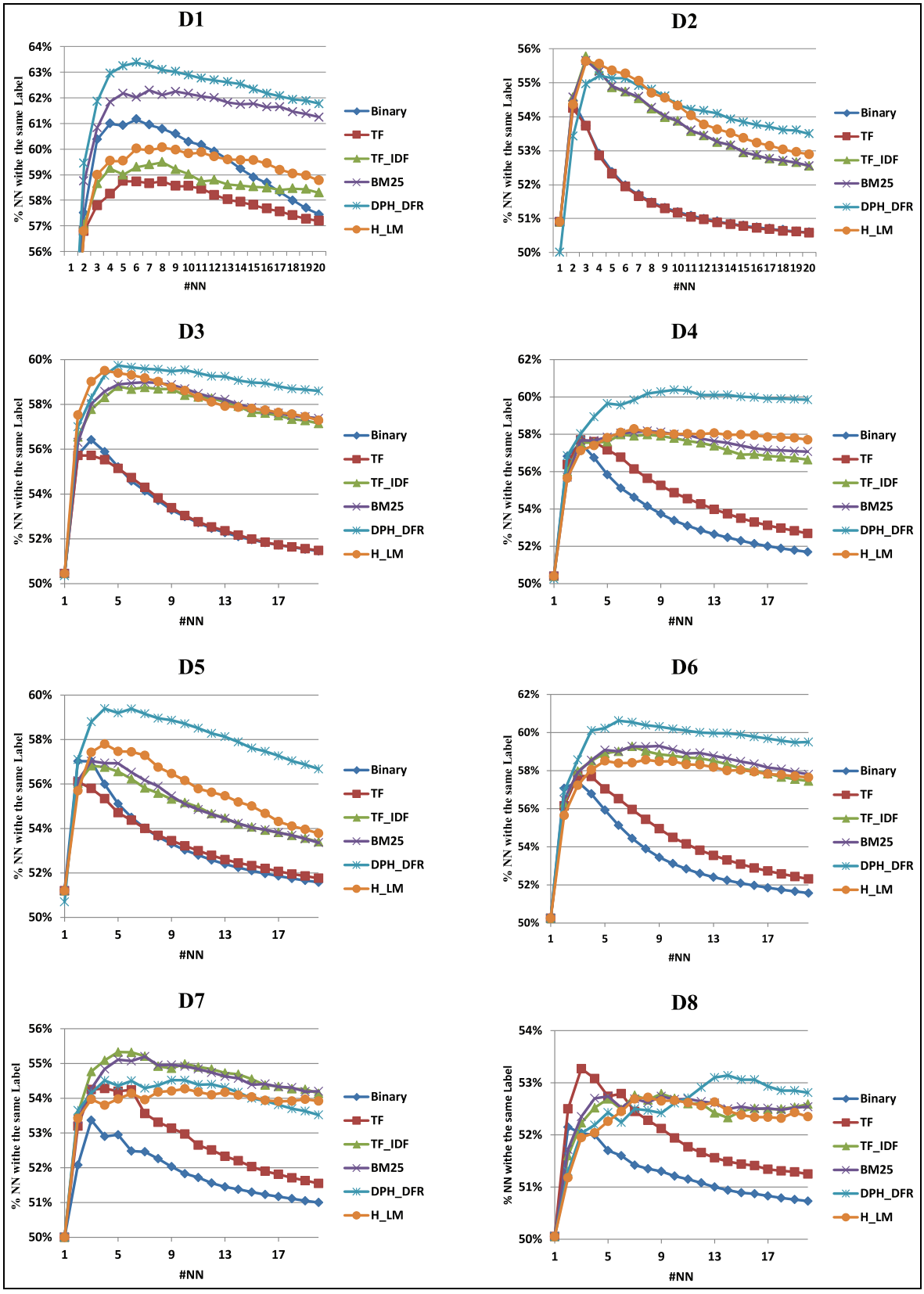

Figure 3. Percentage of *r*-nearest neighbours (using cosine) that share the same label for each dataset. Figure is reproduced in colour in online version.

Table 16. The difference in average accuracy by using each pre-processing strategy by dataset and mean.

Dataset	Adjective and adverb word extraction			Word stemming			Stopword removal		
	1**	0**	Diff.	*1*	*0*	Diff.	**1	**0	Diff.
D1	56.16%	56.26%	−0.17%	56.22%	56.20%	0.04%	56.07%	56.34%	−0.48%
D2	51.33%	52.00%	−1.29%	51.76%	51.58%	0.35%	51.65%	51.69%	−0.09%
D3	52.80%	52.40%	0.76%	52.42%	52.79%	−0.70%	51.99%	53.22%	−2.31%
D4	52.95%	51.68%	2.46%	52.28%	52.36%	−0.16%	52.36%	52.28%	0.16%
D5	54.41%	53.32%	2.06%	53.73%	53.99%	−0.48%	53.66%	54.07%	−0.76%
D6	54.05%	53.43%	1.15%	53.74%	53.74%	−0.01%	54.06%	53.43%	1.18%
D7	63.75%	62.19%	2.52%	63.37%	62.57%	1.27%	63.32%	62.62%	1.12%
D8	57.05%	56.05%	1.77%	56.67%	56.43%	0.42%	56.20%	56.90%	−1.23%
Overall	55.31%	54.67%	1.18%	55.02%	54.96%	0.12%	54.91%	55.07%	−0.28%

Table 17. The difference in the highest accuracy using each pre-processing strategy by dataset and mean.

Dataset	Adjective and adverb word extraction			Word stemming			Stopword removal		
	1**	0**	Diff.	*1*	*0*	Diff.	**1	**0	Diff.
D1	76.21%	69.89%	9.05%	73.14%	72.96%	0.24%	73.19%	72.91%	0.38%
D2	62.43%	70.55%	−11.52%	67.53%	65.45%	3.17%	70.78%	62.20%	13.79%
D3	64.99%	69.38%	−6.32%	66.74%	67.63%	−1.31%	65.33%	69.04%	−5.38%
D4	65.40%	63.19%	3.50%	63.94%	64.65%	−1.10%	64.68%	63.91%	1.19%
D5	71.43%	67.23%	6.25%	68.30%	70.35%	−2.91%	68.96%	69.69%	−1.04%
D6	71.51%	68.68%	4.13%	68.94%	71.25%	−3.25%	71.38%	68.81%	3.72%
D7	91.73%	90.29%	1.59%	91.71%	90.30%	1.56%	90.26%	91.75%	−1.62%
D8	78.28%	72.80%	7.52%	75.88%	75.20%	0.90%	74.53%	76.55%	−2.65%
Overall	72.75%	71.50%	1.74%	72.02%	72.22%	−0.28%	72.39%	71.86%	0.73%

of this analysis with the value of r from 1 to 20. Firstly, considering Binary and TF weighting models, almost for each dataset and r value, we see that these two weighting models yield substantially worse nearest neighbourhoods than other four weighting models, especially as r is increasing. Moreover, for D2, D5 and D7, where TF_IDF performs similar to BM25, it is found that TF_IDF and BM25 possess almost the same quality of nearest neighbourhood as r increases. For D7, TF_IDF weighting produces slightly better nearest neighbourhoods than DPH_DFR and H_LM (with its clustering results being correspondingly a little better than DPH_DFR and H_LM in quality). As all clustering algorithms are more or less dependent on the quality of nearest neighbourhoods, this provides a reasonable explanation for our different by-dataset results.

4.3. Results on data pre-processing strategies

To determine whether each of the three data pre-processing steps is necessary for review clustering, we compared the clustering results on the three data pre-processing steps by collapsing the results of eight different combinations for the substeps, which have been illustrated in three tables: (1) by dataset and mean over all the data pre-processing steps for that dataset (Table 16); (2) by dataset and the best performance of each data pre-processing step for that dataset (Table 17); and (3) by dataset and mean over all the data pre-processing steps on four best clustering algorithms (i.e. *Kmeans, RBR-Kmeans, RBR-H1* and *Direct-H1*) for that dataset (Table 18). The Diff. column for each data pre-processing strategy in Tables 16–18 shows the proportional change of accuracy value by adopting the corresponding pre-processing substep. The overall rows in Tables 16–18 indicate that, on average, the substep of adjective and adverb words extraction can offer some improvements on clustering performance, while stemming and stopword removal strategies indeed bring very limited variation on clustering performance. More specifically, in most cases, the latter two substeps provide even worse results.

In order to explore the influences of three pre-processing strategies on clustering accuracy more clearly, we also conducted statistical significance tests on paired comparisons of clustering results by whether or not adopting each pre-processing strategy over all eight benchmarked datasets (see Table 19). For each pre-processing strategy, there are in total

Table 18. The difference in average accuracy on four best algorithms by using each pre-processing strategy by dataset and mean.

Dataset	Adjective and adverb word extraction			Words stemming			Stopword removal		
	1**	0**	Diff.	*1*	*0*	Diff.	**1	**0	Diff.
D1	69.47%	61.96%	12.11%	65.45%	65.98%	−0.81%	65.86%	65.57%	0.44%
D2	53.20%	54.68%	−2.70%	54.29%	53.59%	1.31%	53.57%	54.31%	−1.35%
D3	58.28%	55.44%	5.11%	56.28%	57.44%	−2.03%	54.98%	58.74%	−6.40%
D4	58.08%	53.47%	8.62%	55.69%	55.86%	−0.29%	55.59%	55.96%	−0.67%
D5	63.60%	59.23%	7.39%	60.87%	61.96%	−1.77%	60.53%	62.29%	−2.83%
D6	61.90%	59.46%	4.10%	60.21%	61.14%	−1.52%	62.03%	59.32%	4.57%
D7	88.90%	82.88%	7.25%	86.01%	85.77%	0.27%	85.48%	86.30%	−0.96%
D8	71.35%	64.65%	10.36%	67.92%	68.08%	−0.23%	67.57%	68.43%	−1.27%
Overall	65.60%	61.47%	6.71%	63.34%	63.73%	−0.61%	63.20%	63.87%	−1.04%

Table 19. Paired comparative results of statistical significance tests for each pre-processing strategy on clustering accuracy.

Dataset	$Accuracy_{1**} > Accuracy_{0**}$		$Accuracy_{*1*} > Accuracy_{*0*}$		$Accuracy_{**1} > Accuracy_{**0}$	
	t-Test value	*p*-value	*t*-Test value	*p*-value	*t*-Test value	*p*-value
D1	−0.245	0.807	−4.287	2.23×10^{-5}***	−4.856	1.68×10^{-6}***
D2	−4.076	5.45×10^{-5}***	1.303	0.193	−0.304	0.762
D3	2.232	0.026*	−2.973	0.003**	−7.951	1.63×10^{-14}***
D4	7.116	4.67×10^{-12}***	−0.697	0.486	0.676	0.499
D5	4.618	5.13×10^{-6}***	−1.826	0.069	−2.535	0.012*
D6	3.171	0.002**	−0.034	0.973	3.452	6.10×10^{-4}***
D7	6.212	1.23×10^{-9}***	0.768	0.443	0.152	0.880
D8	3.392	7.58×10^{-4}***	1.835	0.067	−4.405	1.34×10^{-5}***
Overall	7.810	7.55×10^{-15}***	−2.560	0.011*	−5.406	6.88×10^{-8}***

***$p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

864 ($18 \times 6 \times 8$) paired results of clustering accuracy to make comparisons. One can see that the results in Table 19 are mostly consistent with those in Tables 16–18, implying that adjective and adverb words extraction strategy can offer significant improvements in clustering performance, while adopting stemming and stopword removal strategies would bring distinctly negative influences.

5. Discussion

So far we have conducted comprehensive experiments to show the influence of data pre-processing, term weighting models and clustering algorithms on the performances of online review sentiment clustering. However, the above experiments and experimental results are all based on balanced review datasets, in which the number of positive reviews and that of negative reviews are identical. Although balanced datasets are widely utilized in many research papers about sentiment classification and clustering [7, 18, 19, 22–24, 26–28, 32–34, 37–40], unbalanced data with far fewer reviews for one particular category (e.g. negative reviews) are common in real-life applications (e.g. online product review platforms), which would probably have different effect on the performance of sentiment clustering, especially for the clustering algorithms.

In the literature, *Kmeans*-type algorithms (e.g. *Kmeans, RB-Kmeans, PAM, CLARANS, RBR-H1, Direct-H1* in our experiments) have been proved to tend to reduce variation in cluster sizes if the variation of the 'true' cluster sizes is high [72]. Thus it is possible that *Kmeans*-type algorithms may perform relatively poorly on real-life unbalanced review data, since there is normally a very small proportion of negative reviews (e.g. 20% or less) in reality, while intuitively, the effects of term weighting modelling and pre-processing techniques on sentiment clustering seem not so sensitive to the distribution of review data. In order to have a preliminary understanding about the effect of unbalanced data on sentiment clustering, we provide here several exploratory results of data experiments for future works. For each of D2, D7 and D8, whose size is much larger than 2000, we randomly extracted 1600 positive reviews (i.e. 80%) and 400 negative reviews (i.e. 20%) to form the corresponding unbalanced benchmarked dataset (named D2′, D7′, D8′, respectively). Then for each new dataset, only a pre-processing strategy of adjective and adverb words extraction has been adopted for

Table 20. The percentage difference of average accuracy for each clustering algorithm over the best performance by new unbalanced dataset.

Algorithm	D2′	D7′	D8′	AvgRank
Kmeans	− 20.1	− 13.6	− 18.4	14.0
RB-Kmeans	− 20.2	− 13.6	− 18.4	14.3
PAM	− 9.5	− 15.3	− 15.4	11.0
CLARANS	− 14.2	− 18.3	− 26.2	15.0
Spect-Un	− 1.3	0	0	5.0
Spect-RW	− 0.1	0	0	4.3
Spect-Sy	− 18.2	0	0	7.7
PCA-Kmeans	− 0.4	− 0.9	− 3.2	7.3
NC-NMF	− 25.7	− 15	− 26.3	16.0
UPGMA	0	0	0	4.3
Slink	0	0	0	1.0
Clink	− 15.2	− 15.9	− 13.9	11.3
RBR-H1	− 16.1	− 16.5	− 16.8	14.7
Direct-H1	− 16	− 16.5	− 16.8	14.3
Agglo-I2	0	0	0	4.3
Agglo-H1	− 13.7	− 12.1	− 14.7	9.7
Agglo-WSlink	0	0	0	1.0
Agglo-WClink	− 15.2	− 15.9	− 13.9	11.3

Table 21. The percentage difference of highest accuracy for each clustering algorithm over the best performance by new unbalanced dataset.

Algorithm	D2′	D7′	D8′	AvgRank
Kmeans	− 12.1	− 10.8	− 13	14.3
RB-Kmeans	− 12.1	− 10.8	− 13	14.3
PAM	− 2.3	− 10.3	− 6.2	13.3
CLARANS	− 12.6	− 2.6	− 21	14.7
Spect-Un	− 0.1	− 1	− 0.2	6.0
Spect-RW	0	− 1	− 0.2	5.7
Spect-Sy	− 0.1	− 1	− 0.2	6.0
PCA-Kmeans	− 0.4	− 1	− 0.1	5.3
NC-NMF	− 18.4	− 5.6	− 22	15.7
UPGMA	0	− 0.9	− 0.1	2.0
Slink	0	− 0.9	− 0.1	2.0
Clink	− 1.2	− 6.9	− 3	10.7
RBR-H1	− 12.3	− 9.7	− 13.9	14.7
Direct-H1	− 12.2	− 9.7	− 13.9	15.3
Agglo-I2	0	− 0.9	0	2.3
Agglo-H1	− 0.5	0	− 1.3	6.3
Agglo-WSlink	0	− 0.9	− 0.1	2.0
Agglo-WClink	− 1.2	− 6.9	− 3	10.7

each review, with the six term weighting models and 18 clustering algorithms unchanged. Tables 20 and 21 illustrate the sentiment clustering performances for each clustering algorithm on the new unbalanced datasets and Tables 22 and 23 show the results for term weighting models.

It is noticed that the exploratory experiments on unbalanced datasets show some interesting results. Specifically, the AvgRank columns in Tables 20 and 21 indicate that, on average, *Kmeans*-type algorithms (i.e. *Kmeans, RB-Kmeans, PAM, CLARANS, RBR-H1, Direct-H1*) in our experiments perform rather poorly on clustering accuracy, which is consistent with our assumption above. Meanwhile, some clustering algorithms showing quite poor performances on balanced datasets (e.g. *Agglo-WSlink, Slink, UPGMA, Spect-Sy*, Spect-*RW, PCA-Kmeans, Spect-Un*) do very well on the unbalanced datasets. Moreover, from Tables 22 and 23, we can see that averagely BM25, DPH_DFR and H_LM weighting models are somewhat better than TF_IDF, Binary as well as TF on the new unbalanced datasets, which is still unchanged

Table 22. The percentage difference of average accuracy for each weighting model over the best performance by new unbalanced dataset.

Weighting model	D2′	D7′	D8′	AvgRank
Binary	− 7.8	− 5.3	− 3.9	4.0
TF	− 4.7	− 5.1	− 1.4	3.8
TF_IDF	− 1.3	0	− 4.9	3.7
BM25	0	− 2.1	− 1.4	3.0
DPH_DFR	− 3.1	− 1.1	0	3.1
H_LM	− 1.4	− 4.3	− 0.7	3.3

Table 23. The percentage difference of highest accuracy for each weighting model over the best performance by new unbalanced dataset.

Weighting model	D2′	D7′	D8′	AvgRank
Binary	− 1	− 1	− 0.1	4.3
TF	− 0.6	− 0.9	− 0.1	3.0
TF_IDF	− 0.5	− 0.9	− 0.2	2.7
BM25	0	0	− 0.1	1.3
DPH_DFR	− 0.1	− 0.9	− 0.2	2.5
H_LM	− 0.1	− 0.9	0	2.0

compared with the situation on balanced ones. We can see that, by considering the review distribution, the research framework is needed to redesign and thus the conclusions are needed to recheck.

Therefore, there are considerable future works worth conducting for us to make the answers to the three research questions more robust. For instance, since each of the benchmark datasets of D1, D3, D4, D5 and D6 only contains 1000 positive and 1000 negative reviews, besides D2, D7 and D8, more unbalanced big benchmark review datasets need to be found to conduct the comparative experiments. In addition, on account that many validation measures (e.g. entropy, purity, *F*-measure, etc.) have been developed for evaluating the performance of clustering algorithms [72–74], it is desirable to adopt more evaluation measures, rather than only accuracy. It is obvious that a great deal of intensive work is needed to conduct to solve the above problems.

6. Conclusion

Clustering is a powerful tool for sentiment analysis from texts. As an important unsupervised learning method, it is known that the clustering results may be affected by any step of data pre-processing strategy, VSM model and clustering algorithm in the clustering process.

This paper presents the results of an experimental study of some common clustering techniques with respect to the tasks of sentiment analysis. Different from the previous studies, in particular, we study the combination effects of these factors with a series of comprehensive experimental investigations.

With a collection of very detailed experiments and exploratory trial for future works, we find on average the following experimental conclusions for online review sentiment clustering:

- The *Kmeans*-type clustering algorithms (i.e. *Kmeans, RB-Kmeans, PAM, CLARANS, RBR-H1, Direct-H1*) show clear advantages on balanced review datasets, while performing rather poorly on unbalanced datasets by considering clustering accuracy. Meanwhile, some clustering algorithms showing quite poor performances on balanced datasets (e.g. *Agglo-WSlink, Slink, UPGMA, Spect-Sy, Spect-RW, PCA-Kmeans, Spect-Un*) may do very well on the unbalanced ones.
- The three comparatively newly designed weighting models (e.g. BM25, DPH_DFR and H_LM) are somewhat better than the traditional weighting models (e.g. Binary, TF and TF_IDF) for sentiment clustering on both balanced and unbalanced datasets.

- Adjective and adverb words extraction strategy can offer obvious improvements on clustering performance, while adopting stemming and stopword removal strategies would even bring negative influences on sentiment clustering.

The experiment methods and conclusions would be valuable for both the study and usage of clustering methods in online review sentiment analysis.

Acknowledgements

We give thanks to the anonymous reviewers for their thoughtful comments and suggestions.

Fundings

This work was partly supported by the National Natural Science Foundation of China (71402007/71473143/71271044/U1233118).

Notes

1. http://www.imdb.com/
2. http://www.tripadvisor.com/
3. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

References

[1] Pang B and Lee L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2008; 2(1–2): 1–135.
[2] Pang B, Lee L and Vaithyanathan S. Thumbs up? sentiment classification using machine learning techniques. In: *Proceedings of conference on empirical methods in natural language processing (EMNLP'02)*, Philadelphia, PA, 2002, pp. 79–86.
[3] Prabowo R and Thelwall M. Sentiment analysis: A combined approach. *Journal of Informetrics* 2009; 3(2): 143–157.
[4] Turney PD and Pantel P. From frequency to meaning : Vector space models of semantics. *Journal of Artificial Intelligence Research* 2010; 37(1): 141–188.
[5] Xu W, Liu X and Gong Y. Document clustering based on non-negative matrix factorization. In: *Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval*, Toronto. New York: ACM, 2003, pp. 267–273.
[6] Jain AK, Murty MN and Flynn PJ. Data clustering: A review. *ACM Computing Surveys* 1999; 31(3): 264–323.
[7] Huang X and Croft WB. A unified relevance model for opinion retrieval. In: *Proceedings of the 18th ACM conference on information and knowledge management (CIKM'09)*, Hong Kong. New York: ACM, 2009, pp. 947–956.
[8] Zhong S and Ghosh J. Generative model-based document clustering: A comparative study. *Knowledge and Information Systems* 2005; 8(3): 374–384.
[9] Han J and Kamber M. *Data mining: Concepts and techniques*, 2nd edn. San Francisco, CA: Morgan Kaufmann, 2006.
[10] Liu B. *Sentiment analysis and opinion mining*. San Rafael, CA: Morgan & Claypool, 2012.
[11] Cambria E, Schuller B, Xia Y et al. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems* 2013; (2): 15–21.
[12] Su F and Markert K. From words to senses: A case study of subjectivity recognition. In: *Proceedings of the 22nd international conference on computational linguistics – Volume 1*. Stroudsburg, PA: Association for Computational Linguistics, 2008, pp. 825–832.
[13] Andrzejewski D and Zhu X. Latent Dirichlet allocation with topic-in-set knowledge. In: *Proceedings of the NAACL HLT 2009 workshop on semi-supervised learning for natural language processing*. Stroudsburg, PA: Association for Computational Linguistics, 2009, pp. 43–48.
[14] Zhai Z, Liu B, Xu H et al. Clustering product features for opinion mining. In: *Proceedings of the fourth ACM international conference on web search and data mining*. New York: ACM, 2011, pp. 347–354.
[15] Zhu L, Galstyan A, Cheng J et al. Tripartite graph clustering for dynamic sentiment analysis on social media. In: *Proceedings of the 2014 ACM SIGMOD international conference on management of data*. New York: ACM, 2014, pp. 1531–1542.
[16] Steinbach M, Karypis G and Kumar V. A comparison of document clustering techniques. In: *KDD-2000 workshop on text mining*, Boston, MA, 2000, pp. 525–526.
[17] Karypis G. *CLUTO—software for clustering high-dimensional datasets*, 2007. Available from: http://www.cs.umn.edu/~cluto
[18] Agarwal R, Prabhakar T and Chakrabarty S. 'I know what you feel': Analyzing the role of conjunctions in automatic sentiment analysis. In: Nordström B and Ranta A (eds), *Proceedings of the 6th internationl conference on advances in natural language processing*, Gothenburg. Berlin: Springer, 2008, pp. 28–39.

[19] Argamon S, Whitelaw C, Chase P et al. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology* 2007; 58(6): 802–822.
[20] Baccianella S, Esuli A and Sebastiani F. Multi-facet rating of product reviews. In: Boughanem M, Berrut C, Mothe J and Soule-Dupuy C (eds), *Proceedings of the 31st european conference on information retrieval (ECIR'09)*, Toulouse. Berlin: Springer, 2009, pp. 461–472.
[21] Baccianella S, Esuli A and Sebastiani F. Feature selection for ordinal regression. In: *Proceedings of the 2010 ACM symposium on applied computing (SAC'10)*, Sierre, Switzerland. New York: ACM, 2010, pp. 1748–1754.
[22] Blitzer J, Crammer K, Kulesza A et al. Learning bounds for domain adaptation. In: *Proceedings of the 22nd annual conference on neural information processing systems*, Vancouver, 2008, pp. 129–136.
[23] Blitzer J, Dredze M and Pereira F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: *Proceedings of the 45th annual meeting of the Association for Computational Linguistics (ACL'07)*, Prague, 2007, pp. 440–447.
[24] Boiy E and Moens M-F. A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval* 2009; 12(5): 526–558.
[25] Dredze M and Crammer K. Online methods for multi-domain learning and adaptation. In: *Proceedings of the conference on empirical methods in natural language processing*, Honolulu, HI. Stroudsburg, PA: Association for Computational Linguistics, 2008, pp. 689–697.
[26] Dredze M, Crammer K and Pereira F. Confidence-weighted linear classification. In: *Proceedings of the 25th international conference on machine learning (ICML'08)*, Helsinki. New York: ACM, 2008, pp. 264–271.
[27] Gao S and Li H. A cross-domain adaptation method for sentiment classification using probabilistic latent analysis. In: *Proceedings of the 20th ACM international conference on information and knowledge management (CIKM'11)*, Glasgow. New York: ACM, 2011, pp. 1047–1052.
[28] Goldberg AB and Zhu X. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In: *Proceedings of the first workshop on graph based methods for natural language processing*. Stroudsburg, PA: Association for Computational Linguistics, 2006, pp. 45–52.
[29] Jindal N and Liu B. Review spam detection. In: *Proceedings of the 16th international conference on World Wide Web*, Banff, Alberta. New York: ACM, 2007, pp. 1189–1190.
[30] Jindal N and Liu B. Opinion spam and analysis. In: *Proceedings of the international conference on Web search and web data mining (WSDM'08)*, Palo Alto, CA. New York: ACM, 2008, pp. 219–230.
[31] Jindal N, Liu B and Lim E-P. Finding unusual review patterns using unexpected rules. In: *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM'10)*, Toronto, ON. New York: ACM, 2010, pp. 1549–1552.
[32] Li G and Liu F. Application of a clustering method on sentiment analysis. *Journal of Information Science* 2012; 38(2): 127–139.
[33] Li T, Sindhwani V, Ding C et al. Knowledge transformation for cross-domain sentiment classification. In: *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, Boston, MA. New York: ACM, 2009, pp. 716–717.
[34] Mansour Y, Mohri M and Rostamizadeh A. Domain adaptation with multiple sources. In: *Proceedings of the 23rd annual conference on neural information processing systems*, Vancouver, 2009, pp. 1041–1048.
[35] Mukherjee A, Liu B and Glance N. Spotting fake reviewer groups in consumer reviews. In: *Proceedings of the 21st international conference on World Wide Web*, Lyon. New York: ACM, 2012, pp. 191–200.
[36] Pan SJ, Ni X, Sun J-T et al. Cross-domain sentiment classification via spectral feature alignment. In: *Proceedings of the 19th international conference on World Wide Web*, Raleigh, NC. New York: ACM, 2010, pp. 751–760.
[37] Pang B and Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: *Proceedings of the 42nd annual meeting of the Association for Computational Linguistics*, Barcelona. Stroudsburg, PA: Association for Computational Linguistics, 2004.
[38] Pang B and Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics*, Ann Arbor, MI. Stroudsburg, PA: Association for Computational Linguistics, 2005, pp. 115–124.
[39] Sindhwani V and Melville P. Document-word co-regularization for semi-supervised sentiment analysis. In: *Eighth IEEE international conference on data mining (ICDM '08)*, 2008, pp. 1025–1030.
[40] Venkatasubramanian S, Veilumuthu A, Krishnamurthy A et al. A non-syntactic approach for text sentiment classification with stopwords. In: *Proceedings of the 20th international conference companion on World Wide Web*, Hyderabad. New York: ACM, 2011, pp. 137–138.
[41] Toutanova K and Manning CD. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora (EMNLP/VLC-2000)*, Hong Kong, 2000, pp. 63–70.
[42] Porter AA and Selby RW. Empirically guided software development using metric-based classification trees. *IEEE Software* 1990; 7(2): 46–54.

[43] Salton G. *The SMART retrieval system – Experiments in automatic document processing*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[44] Salton G, Buckley C and Fox EA. Automatic query formulations in information retrieval. *Journal of the American Society for Information Science* 1983; 34(4): 262–280.
[45] Salton G, Fox EA and Wu H. Extended Boolean information retrieval. *Communications of the ACM* 1983; 26(11): 1022–1036.
[46] Baeza-Yates R and Ribeiro-Neto B. *Modern information retrieval*. New York: Addison-Wesley, 1999.
[47] Salton G, Wu H and Yu CT. The measurement of term importance in automatic indexing. *Journal of the American Society for Information Science* 1981; 32(3): 175–186.
[48] Jones KS. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 1972; 28(1): 11–21.
[49] Robertson SE and Jones KS. Relevance weighting of search terms. *Journal of the American Society for Information Science* 1976; 27(3): 129–146.
[50] Robertson SE, Walker S, Jones S et al. Okapi at TREC-3. In: *The third text retrieval conference (TREC '94)*, 1994.
[51] Amati G, Ambrosi E, Bianchi M et al. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In: *Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.
[52] Hiemstra D. *Using language models for information retrieval*. PhD thesis, University of Twente, 2001.
[53] Santos R, Macdonald C and Ounis L. Exploiting query reformulations for web search result diversification. In: *Proceedings of the 19th international conference on World Wide Web*, Raleigh, NC. New York: ACM, 2010, pp. 881–890.
[54] Zhao Y and Karypis G. *Criterion functions for document clustering: Experiments and analysis*. University of Minnesota, Department of Computer Science/Army HPC Research Center, 2001.
[55] Zhao Y and Karypis G. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery* 2005; 10(2): 141–168.
[56] Zhao Y and Karypis G. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning* 2004; 55(3): 311–331.
[57] Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 1982; 28(2): 129–137.
[58] Kaufman L and Rousseeuw PJ. *Finding groups in data: An introduction to cluster analysis*. New York: Wiley Online Library, 1990.
[59] Ng RT and Han J. CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering* 2002; 14(5): 1003–1016.
[60] Luxburg Uv. A tutorial on spectral clustering. *Statistics and Computing* 2007; 17(4): 395–416.
[61] Shi J and Malik J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000; 22(8): 888–905.
[62] Ng AY, Jordan MI and Weiss Y. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2001, pp. 849–856.
[63] Pearson K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 1901; 2559–572.
[64] Beil F, Ester M and Xu X. Frequent term-based text clustering. In: *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining*, Edmonton, Alberta. New York: ACM, 2002, pp. 436–442.
[65] Fung BCM, Wang K and Ester M. Hierarchical document clustering using frequent itemsets. In: *Proceedings of the SIAM international conference on data mining*, San Francisco, CA, 2003, pp. 59–70.
[66] Slonim N and Tishby N. Document clustering using word clusters via the information bottleneck method. In: *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval*, Athens. New York: ACM, 2000, pp. 208–215.
[67] Chehreghani MH, Abolhassani H and Chehreghani MH. Density link-based methods for clustering web pages. *Decision Support Systems* 2009; 47(4): 374–382.
[68] Hofmann T. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning* 2001; 42(1): 177–196.
[69] Luo C, Li Y and Chung SM. Text document clustering based on neighbors. *Data & Knowledge Engineering* 2009; 68(11): 1271–1288.
[70] Zhong S. Semi-supervised model-based document clustering: A comparative study. *Machine Learning* 2006; 65(1): 3–29.
[71] Peng J. *Learning to select for information retrieval*. PhD thesis, University of Glasgow, 2010.
[72] Xiong H, Wu JJ and Chen J. *K*-Means clustering versus validation measures: A data-distribution perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 2009; 39(2): 318–331.
[73] Aliguliyev RM. Performance evaluation of density-based clustering methods. *Information Sciences* 2009; 179(20): 3583–3602.
[74] Vinh NX, Epps J and Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research* 2010; 11: 2837–2854.