Pattern Recognition 76 (2018) 691–703

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Concept decompositions for short text clustering by identifying word communities

Caiyan Jia[a,*], Matthew B. Carson[b], Xiaoyang Wang[a], Jian Yu[a]

[a] School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China
[b] Division of Health and Biomedical Informatics, Department of Preventive Medicine, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611, USA

ARTICLE INFO

Article history:
Received 11 July 2016
Revised 23 August 2017
Accepted 30 September 2017
Available online 10 October 2017

MSC:
00-01
99-00

Keywords:
Short text clustering
Concept decomposition
Spherical *k*-means
Semantic word community
Community detection

ABSTRACT

Short text clustering is an increasingly important methodology but faces the challenges of sparsity and high-dimensionality of text data. Previous concept decomposition methods have obtained concept vectors via the centroids of clusters using *k*-means-type clustering algorithms on normal, full texts. In this study, we propose a new concept decomposition method that creates concept vectors by identifying semantic word communities from a weighted word co-occurrence network extracted from a short text corpus or a subset thereof. The cluster memberships of short texts are then estimated by mapping the original short texts to the learned semantic concept vectors. The proposed method is not only robust to the sparsity of short text corpora but also overcomes the curse of dimensionality, scaling to a large number of short text inputs due to the concept vectors being obtained from term-term instead of document-term space. Experimental tests have shown that the proposed method outperforms state-of-the-art algorithms.

1. Introduction

In the current Web 2.0 era, an increasing number of short texts have been generated including search result snippets, forum titles, image or video titles and tags, frequently asked questions, tweets, microblogs, and so on. This has resulted in a growing need for fast and efficient clustering of short texts according to high similarities within and dissimilarities between clusters. A well-designed short text clustering algorithm has the ability to greatly stimulate and promote its real applications such as topic detection, answering service recommendations, image or video tagging, information retrieval, etc. However, unlike normal texts, use of short texts is complicated by sparsity and high dimensionality. As a result, the classical *tf-idf* (term frequency-inverse document frequency) measure, the vector space model (VSM), and normal text clustering methods may not work well when applied to short texts.

One way to solve the sparsity issue of feature vectors is to expand short texts to long texts by the use of external knowledge sources such as Wikipedia [1], WordNet [2], HowNet [3], Web search results [4], other user constructed knowledge bases [5–7], peripheral information sources [8,9], etc. However, these external knowledge-enhanced methods have the following two issues. First, the creation and maintenance of such resources (e.g., Wikipedia and WordNet) can be very expensive. Second, this introduces the new challenge of how to properly use those external resources (see [10] as an example). In most cases, solving this new problem itself is time-consuming and complicated.

Alternatively, some research efforts have concentrated on improving traditional methods of normal text clustering or designing new models to handle short texts. Existing methods include probabilistic topic models (BTM [11] and GSDMM [12]), and heuristic optimization methods (nonnegative matrix factorization methods [13,14], extended vector space models [15,16], and other heuristic and search snippet specific methods [17–19]). The main idea of these methods such as BTM, TNMF [14], and Generalized VSM [15] is to make use of the relationships between pairs of terms in order to compensate for the sparsity of short texts. However, these methods ignore the relationships among three or more terms. As evidenced by the psychologist's statement "Concepts are the glue that holds our mental world together" [20], words related to the same topic are likely to co-occur in the same text, thus they link together and form densely connected communities in the word co-occurrence network of a corpus. Therefore, it is possible to ex-

[*] Corresponding author.
E-mail address: cyjia@bjtu.edu.cn (C. Jia).

https://doi.org/10.1016/j.patcog.2017.09.045
0031-3203/© 2017 Elsevier Ltd. All rights reserved.

tract concept vectors directly from the word co-occurrence network rather than from the document-term space (the classical method of the latter is spherical k-means [21]). The effect can be seen in methods such as conceptual grouping [22] and word sense induction [23], both of which concentrate on fine-grained word clusters and have not been successfully used to cluster short texts to the best of our knowledge.

In addition, short texts are very sparse, thus their terms are particularly valuable. For example, corpora with even millions of short texts may only contain a few thousand terms that characterize them. Accordingly, extracting concept vectors from term-term space is beneficial for overcoming the curse of dimensionality for large-scale corpora. Therefore, inspired by the concept decomposition method spherical k-means [21] in normal text clustering, we propose a novel concept decomposition method, WordCom, which is based on the identification of semantic word communities using a k-means-type community detection method.

The procedure of WordCom has four steps. First, we construct the word co-occurrence network for a corpus. Second, we extract the semantic word communities from the network using k-means-type algorithm k-rank-D [24]. In k-rank-D, the initial cluster centers and the number of clusters are determined by actively selecting k potential centers located in the right upper part of the decision graph, which characterizes the likelihood of data points being cluster centers by a higher density than their neighbors and by a relatively large distance from points with higher densities [25]. Third, we combine the word communities and their corresponding centers to form concept vectors. Finally, we project all short texts into these concept vectors and obtain their cluster memberships. Moreover, for a small subset of a large-scale short text corpus, words in the subset may have already covered most of words in the whole corpus. This allows us to obtain the concept centers of the corpus only from its subset, which makes our proposed concept decomposition method scale easily for very large short text corpora.

The remainder of this paper is organized as follows. Section 2 introduces the background of this study including related studies and the basic idea of concept decomposition. Section 3 presents our new proposed concept decomposition method, WordCom. Section 4 shows the comparison results. Section 5 draws conclusions and further considerations.

2. Background

2.1. From normal text to short text clustering

Probabilistic topic models such as PLSA (Probabilistic Latent Semantic Analysis) [26] and LDA (Latent Dirichlet Allocation) [27] are classical methods for uncovering hidden topics from normal text corpora. In these models, both term distribution within topics and topic distribution in texts can be inferred by maximum likelihood estimation methods. To address the sparseness of short texts using the LDA topic model, the BTM (biterm topic model) has been developed to capture term co-occurrence pattern implied in short texts by extending the traditional unigram LDA model to 2-gram LDA model [11]. Subsequently, another variation of LDA, GSDMM (a collapsed Gibbs sampling algorithm of Dirichlet Multinomial Mixture), was proposed to cluster short text corpora and also showed good performance on normal text clustering [12].

Similar to topic models, non-negative matrix factorization (NMF) has the ability to identify hidden structures of terms and texts on topics represented by two matrix factors: term-topic matrix **U** and topic-document matrix **V** [28]. Ncut (a similarity weighted NMF) was developed to tackle the sparsity issue and cluster short text corpora [13]. TNMF (a two-step NMF framework) [14], was proposed later. This method performs symmetric NFM on a term similarity matrix to define a term-topic matrix **U**, followed by inference of the topic-document matrix **V** using the NMF framework on the original term-document matrix X according to the learned **U**.

The vector space model (VSM) is a classical model for representing normal texts. VSM assumes that terms are independent and it ignores the semantic relationships among terms. Therefore, in order to overcome the sparseness of terms in a short text corpus, a generalized VSM [15] is used to represent short texts, where the correlations between pairs of terms are used rather than the weights of single terms. Similarly, the method in [16] uses the group of related keywords extracted from the processed short texts themselves to expand the short text corpus and alleviate its sparsity.

TermCut is a core term-based bisect clustering method [17] for short text clustering. It finds the best term by optimizing the clustering criterion RMcut at each step of bisection. Therefore, at each bisection, it must traverse all remaining terms to find the 'core term'. The high dimensionality of short texts results in a high level of time complexity for TermCut.

2.2. Concept decomposition and spherical k-means

Dhillon et al. [21] have proposed a concept decomposition method named spherical k-means to cluster normal texts. Concept decomposition was later extended to concept factorization [29–31]. In this section, we will introduce the basic idea of concept decomposition.

Let $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ be the text vectors of a corpus. Each text is represented by a vector ($\mathbf{x_i}$, $i = 1, \ldots, n$) of m possible terms x_{ij}, $j = 1, \ldots, m$, where each element of the vector tracks the weight of a term in the text. Usually the weight is measured by the *tf-idf* value, i.e., $x_{ij} = tf_{ij} \times \log(\frac{n}{nd_j})$, where tf_{ij} indicates the frequency of term j in the text i and nd_j denotes the total number of texts containing term j.

Given any two unit vectors **x** and **y** in R^m, the cosine similarity of **x** and **y** is defined by the inner product $\mathbf{x}^T\mathbf{y}$, i.e.,

$$\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|\,\|\mathbf{y}\| cos(\theta(\mathbf{x}, \mathbf{y})) = cos(\theta(\mathbf{x}, \mathbf{y})), \tag{1}$$

where $0 \le \theta(\mathbf{x}, \mathbf{y}) \le \pi/2$ denotes the angle of vectors **x** and **y**.

Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$ denote a partitioning of the text vectors into k disjoint clusters such that

$$\bigcup_{k'=1}^{k} \pi_{k'} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}, \tag{2}$$

$$\pi_r \bigcap \pi_s = \phi, \forall r, s \in \Pi, r \neq s. \tag{3}$$

For each $k' \in 1, 2, \ldots, k$, the centroid of the text vectors contained in the cluster $\pi_{k'}$ is

$$\mathbf{m}_{k'} = \frac{1}{n_{k'}} \sum_{\mathbf{x} \in \pi_{k'}} \mathbf{x}, \tag{4}$$

where $n_{k'}$ is the number of text vectors in $\pi_{k'}$. Then, the concept vector of the cluster $\pi_{k'}$ is defined as

$$\mathbf{c}_{k'} = \frac{\mathbf{m}_{k'}}{\|\mathbf{m}_{k'}\|}. \tag{5}$$

According to Cauchy–Schwarz inequality

$$\sum_{\mathbf{x}\in\pi_{k'}} \mathbf{x}^T\mathbf{z} \le \sum_{\mathbf{x}\in\pi_{k'}} \mathbf{x}^T\mathbf{c}_{k'}, \forall \mathbf{z} \in R^m, \tag{6}$$

to achieve sufficient quality of partitioning Π, spherical k-means maximizes the following objective function

$$f(\{\pi_{k'}, \mathbf{c}_{k'}\}_{k'=1}^{k}) = \sum_{k'=1}^{k} \sum_{\mathbf{x}\in\pi_{k'}} \mathbf{x}^T\mathbf{c}_{k'}. \tag{7}$$

The details of the algorithm are showed in the Algorithm 1 [21]. The spherical k-means is a classic convergence algorithm which either stops when values change by less than ϵ, or after T iterations [32].

It has been observed that concept vectors tend towards 'orthonormality', provide compact summary of the clusters [21], and top weighted words in the concept vector $c_{k'}$ represent the concept or semantics of the cluster $\pi_{k'}$. Therefore, like topic models and NMF methods, spherical k-means is not only able to capture the topic structure of the processed texts but can also obtain the term distribution under each topic.

The spherical k-means algorithm is a variant of the well known Euclidean k-means algorithm, and the time complexity of spherical k-means is $O(Tnmk)$. However, spherical k-means is more interpretable than k-means because each text vector can be approximated by a linear combination of concept vectors in a low rank k-dimensional space.

3. WordCom: Concept decompositions using word communities

Each short text within a corpus contains few words and the vast majority occur only once. This makes term frequency useless in the *tf-idf* measurement. Moreover, if a VSM representation is used on a short text corpus, the sparse and high-dimensional vectors will result in a waste of both memory and computation time [12]. Not only do terms occurring in the same text imply the same semantics, but terms appearing in different texts with a common co-occurring term may have the same semantics as well. Taking two short texts, 'apple fruit' and 'pear fruit', as examples, 'apple' and 'pear' do not co-occur in the same text but share a common co-occurring word, 'fruit'. Thus, they may belong to the same concept. That is to say, words in the same concept tend to connect densely in the word co-occurrence network, with an edge between two words if they appear in the same short text.

Therefore, our first step was to construct the word co-occurrence network for a short text corpus to characterize the relationships among words. We then used the k-means-type algorithm k-rank-D [24] to identify semantic word communities in the word co-occurrence network while extracting community centers (we used 'communities' instead of 'clusters' in the network scenario). Next, we combined the word communities with the extracted centers to form concept vectors implied in term-term space. Finally,

Algorithm 1: Spherical k-means algorithm.

```
Data: The $n \times m$ matrix $\mathbf{X}$ of a corpus.
Result: The partitioning $\Pi$ of $\mathbf{X}$.
Initialize a partitioning of $\mathbf{X}$: $\{\pi_{k'}^{(0)}\}_{k'=1}^{k}$ randomly, then compute the
concept vectors $\{\mathbf{c}_{k'}^{(0)}\}_{k'=1}^{k}$; set $t = 0$, the maximum iteration number be
$T$, error rate $\epsilon > 0$;
while $t < T$ do
    forall the $1 \le i \le n$ do
        Find the concept vector $\mathbf{c}_{k'}$ which has maximal cosine similarity
        to $\mathbf{x}_i$ and put $\mathbf{x}_i$ to $\pi_{k'}^{(t+1)}$, where
            $\pi_{k'}^{(t+1)} = \{\mathbf{x}_i \in \mathbf{X} : \mathbf{x}_i^T\mathbf{c}_{k'}^{(t)} > \mathbf{x}_i^T\mathbf{c}_l^{(t)}, 1 \le l \le k, l \ne k'\};$    (8)
    end
    forall the $1 \le k' \le k$ do
        Update the concept vectors
            $\mathbf{c}_{k'}^{(t+1)} = \frac{\mathbf{m}_{k'}^{(t+1)}}{\|\mathbf{m}_{k'}^{(t+1)}\|};$    (9)
    end
    if $\|f(\{\pi_{k'}^{(t+1)}, \mathbf{c}_{k'}^{(t+1)}\}_{k'=1}^{k}) - f(\{\pi_{k'}^{(t)}, \mathbf{c}_{k'}^{(t)}\}_{k'=1}^{k})\| < \epsilon$ then
        break;
    end
    $t = t + 1;$
end
```

we projected all short texts by computing the cosine similarity between short texts and concept vectors to obtain the cluster membership indicators of these short texts. The WordCom algorithm was composed of four steps: co-occurrence network construction, semantic word community detection, concept vector formulation, and cluster membership assignment.

3.1. Co-occurrence network construction

Suppose that $G = (V, E)$ is a network, where V is a set of nodes ($\|V\| = N$), E is an edge set that indicates relationships between pairs of nodes ($\|E\| = M$) and is usually represented by an adjacency matrix $A = [A_{uv}]$, where $A_{uv} = w_{uv}$ is the weight of the edge from node u to v, which is 0 if the edge does not exist.

We add an edge between words u and v if and only if they occur in the same short text for constructing the word co-occurrence network. To measure the link strength of words u and v, we use the positive pointwise mutual information [14] as follows.

$$w_{uv} = max\left(log\frac{p(t_u, t_v)}{p(t_u) \times p(t_v)}, 0 \right), \tag{10}$$

where

$$p(t_u, t_v) = \frac{n(t_u, t_v)}{\sum_{w,l} n(t_w, t_l)}, \; p(t_u) = \frac{\sum_w n(t_u, t_w)}{\sum_{w,l} n(t_w, t_l)}, \tag{11}$$

and $n(t_u,\ t_v)$ is the co-occurrence frequency of words u and v. Therefore, for a short text corpus with n short texts and m words, the constructed word co-occurrence network has $N = m$ nodes, $M \le m^2$ edges.

The time complexity is $O(max(nl_{max}^2, m^2))$ for constructing this network, where l_{max} is the maximum length of short texts. For a large-scale short text corpus, $n > > m$ and l_{max} can be bounded by a small constant. Thus, the time complexity is linear with n and quadratic with m at this step.

3.2. Semantic word community detection

A community is a subgraph of a network whose vertices are more tightly connected with each other in the subgraph than with vertices outside the subgraph. Community detection involves grouping the nodes of a network into communities of densely connected nodes. In the literature, numerous methods have been de-

Algorithm 2: k-rank-D for word community detection.

```
Data: The weighted word co-occurrence network of a corpus: $A = [w_{uv}]$.
Result: Word communities $[\mathbf{com}_{k'}]_{k'=1}^{k}$, community centers $[\mathbf{cen}_{k'}]_{k'=1}^{k}$
        and PageRank centrality vector of all nodes $\mathbf{v} = [v_u]_{u=1}^{m}$.
Initialize $\tau = 0.15$, $k$; Let $P = [\frac{w_{uv}}{\sum_v w_{uv}}]$, $I$ be identity matrix, and $\mathbf{v}^0$ be
all-ones vector; Set $S = (A + I)^3$ and normalize $S$ to 1;
1. Calculate PageRank centrality vector $\mathbf{v} = [v_u]_{u=1}^{m}$ using the power method:
   repeat
      $\mathbf{v}^{t+1} = ((1 - \tau)P + I\frac{\tau}{m})\mathbf{v}^t;$   (12)
      Normalize $\mathbf{v}^t$ to 1;
      $t = t + 1;$
   until convergence;
2. Compute the structure distance $d_{uv}$ for each pair of nodes $u$ and $v$;
   Let $\bar{\delta}_v = max_{u \neq v}(d_{uv})$ for the node $v$ with highest PageRank centrality;
   forall the nodes $1 \le u \le m$ do
      Calculate the dispersion of a node $u$ to other nodes with higher
      PageRank centrality by $\bar{\delta}_u = min_{v: v_v > v_u}(d_{uv})$;
   end
3. Draw the decision graph for all $1 \le u \le m$ nodes, where the value of the
   $x$-coordinate is $v_u$ and that of the $y$-coordinate is $\bar{\delta}_u$;
4. Select $k$ initial centers in the right upper part of the decision graph (nodes
   with large $v_u$ and $\bar{\delta}_u$) manually or automatically for $k$ nodes with the
   largest comprehensive value $CV(u)$:
      $CV(u) = v_u \cdot \bar{\delta}_u / (max_{v=1}^{m}(v_v) \cdot max_{v=1}^{m}(\bar{\delta}_v));$   (13)
5. Run the $k$-means algorithm on data vectors $S$ with the $k$ selected initial
   centers.
```

veloped to identify community structure in a complex network [33,34].

In this study, we use k-rank-D [24] to uncover word communities in the constructed word co-occurrence network because it is a k-means type community detection algorithm. Besides detecting word communities, the algorithm identifies word community centers and the importance of nodes measured by PageRank centrality [35]. k-rank-D is actually an improved version of the Signal [36] and k-rank [37] algorithms used for community detection. The difference lies in how initial centers are selected and whether the number of communities is specified. In Signal, the initial centers of communities are randomly generated and the number of communities are supplied in advance. In k-rank, the initial centers are selected by computing their PageRank centrality and the distance between centers, but the number of communities must be specified here as well. In k-rank-D, the initial centers and the number of communities are selected actively from the decision graph of a network. Therefore, k-rank-D is able to avoid a poor choice of initial centers of k-means type algorithms and gives a visualized method to choose initial centers and the number of communities.

Supposing that $A = [w_{uv}]$ is the weighted word co-occurrence network of a corpus, the main procedures of the k-rank-D algorithm for detecting word communities are showed in the Algorithm 2, where the structure distance between two nodes d_{uv} can be computed by transformation of any vertex similarity metric [36,38], τ is the re-start probability in PageRank (fixed at 0.15 in a common setting [35]).

After executing the k-rank-D algorithm on the word co-occurrence network, we obtain a PageRank centrality vector $\mathbf{v} = [v_u]_{u=1}^m$ and word communities denoted by $[\mathbf{com}_{k'}]_{k'=1}^k$, where $\mathbf{com}_{k'} = [com_{k'u}]_{u=1}^m$ (if word u belongs to the community k', then $com_{k'u} = 1$; otherwise $com_{k'u} = 0$; $u = \{1, 2, \ldots, m\}$). The resulting m-dimension cluster centers of data matrix S are given by $[\mathbf{cen}_{k'}]_{k'=1}^k$. The time complexity of k-rank-D is dominated by computing vertex similarity for obtaining initial centers. In the worst case, it is bounded by $O(m^3)$.

3.3. Concept vector formulation

Intuitively, both $[\mathbf{com}_{k'}]_{k'=1}^k$ and $[\mathbf{cen}_{k'}]_{k'=1}^k$ can be concept vectors. However, maximum benefit is obtained by using both of them. The following two reasons explain why choosing this route is beneficial. First, k vectors $[\mathbf{com}_{k'}]_{k'=1}^k$ are completely orthogonal since m words are divided into k disjoint groups. However, in some cases, a word will belong to multiple communities and take on different meanings. For example, in the two short texts ‘apple computer’ and ‘apple fruit’, ‘apple’ is the key word of the topic ‘computer’ as well as the key words of topic ‘fruit’. In addition, each word plays a different role in its respective community, while $[\mathbf{com}_{k'}]_{k'=1}^k$ ignores the topological importance of words $\mathbf{v}$ in their communities. Therefore, using $[\mathbf{com}_{k'}]_{k'=1}^k$ as the sole concept vectors may lead to poor clustering results. Second, although $[\mathbf{cen}_{k'}]_{k'=1}^k$ enables capture of the overlapping property of words in different communities, they do not take the centrality of words in the network into account. Furthermore, we have observed that using only $[\mathbf{cen}_{k'}]_{k'=1}^k$ as concept vectors tends to generate empty clusters since some concept vectors may have strong relevance to the others. Therefore, it may be a good idea to combine these two sets of vectors to form concept vectors.

Thus, we formulate the concept vector $\bar{\mathbf{c}}_{k'}, k' = 1, 2, \ldots, k$ as follows:

$$\bar{\mathbf{c}}_{k'} = \beta\mathbf{com\_w}_{k'} + (1-\beta)\mathbf{cen}_{k'}, k' = 1, 2, \ldots, k, \tag{14}$$

where $[\mathbf{com\_w}_{k'}]_{k'=1}^k$ is the weighted version of $[\mathbf{com}_{k'}]_{k'=1}^k$ (if $com_{k'u} = 1$, then $com\_w_{k'u} = (1-\alpha)com_{k'u} + \alpha v_u$; if $com_{k'u} = 0$, then $com\_w_{k'u} = 0$; $u = \{1, 2, \ldots, m\}$), α balances the importance of community structure and PageRank centrality, β weights the importance of word communities and cluster centers, and α, $\beta \in [0, 1]$. The time complexity of this step is $O(m)$.

3.4. Cluster membership assignment

By using the concept decomposition method with spherical k-means, it is straightforward to project the short text corpus $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ onto the concept vectors $[\bar{\mathbf{c}}_{k'}]_{k'=1}^k$. For each text vector $\mathbf{x}_i$, $1 \le i \le n$, we find the concept vector $\bar{\mathbf{c}}_{k'}$ closest in cosine similarity to $\mathbf{x}_i$ and put $\mathbf{x}_i$ into $\pi_{k'}$, i.e.,

$$\pi_{k'} = \{\mathbf{x}_i \in \mathbf{X} : \mathbf{x}_i^T\bar{\mathbf{c}}_{k'} > \mathbf{x}_i^T\bar{\mathbf{c}}_l, 1 \le l \le k, l \ne k'\}, 1 \le k' \le k. \tag{15}$$

The time complexity of this step is $O(nmk)$.

3.5. Wordcom_s: Wordcom with sampling strategy

To summarize, our proposed concept decomposition method WordCom is composed of the above-described four steps. Taking into account the time complexity of each step, the total time complexity of WordCom is bounded by $O(max\{nmk, m^3\})$. Since short texts are very sparse, the set of words for representing a short text corpus often number in the thousands. It is not difficult for k-rank-D to partition a network with thousands of nodes [24]. Furthermore, the time complexity of k-rank-D is dominated by selecting initial centers. This complexity can be dramatically reduced if we only select initial centers from a subset of nodes, since the initial centers are not necessarily the real centers (i.e., it is enough if they could scatter in different clusters). We will explore this in the future on account of the medium size of word co-occurrence networks encountered in this study.

Alternatively, to further accelerate WordCom on a large-scale corpus, we can extract a subset of the corpus, find concept vectors from the word co-occurrence network of the subset (since the words in the subset may cover most of words in the original corpus), and then project the whole corpus onto the concept vectors to obtain their cluster indicators. We call the version of WordCom using this sampling strategy WordCom_s. The details of algorithm WordCom_s are showed in the Algorithm 3.

Algorithm 3: WordCom_s algorithm.

```
Data: The $n \times m$ matrix $X$ of a corpus.
Result: The partitioning $\Pi$ of $X$.
Initialize sampling ratio $\rho$, the number of word communities $k$;
  1. Sample a subset of a given short text corpus with sampling ratio $\rho$;
  2. Construct the weighted word co-occurrence network for the subset by Eq. (10);
  3. Detect communities in the network by Algorithm 2;
  4. Form concept vectors using Eq. (14);
  5. Extract a new representation of the original corpus using the words contained in the subset;
  6. Assign each short text to its closest concept vector using Eq. (15).
```

WordCom_s has two more steps: steps 1 and 5 (sampling and re-representation) than WordCom. These two steps both operate in linear time with m and n.

4. Performance evaluations and comparisons

In this section, we report an empirical study of the WordCom algorithm compared to state-of-the-art algorithms on several real-

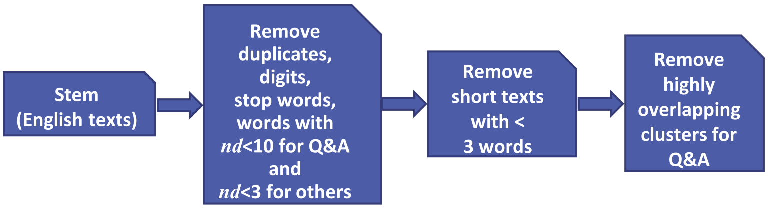


Fig. 1. The workflow for preprocessing short text corpora.

Table 1
Basic properties of the short text corpora.

	Title	Tweet	Tweet_10	Tset	T_50	Q&A	SMS
n	2630	2472	2313	11104	9190	93536	2995
m	1403	4336	4045	6699	5780	4966	1220
k	9	89	55	152	70	30	2
l_{max}	18	31	28	24	22	20	63
l_{ave}	6.24	12.94	12.44	10.2	9.85	4.64	13.99

n and m are the total number of short texts and that of words, k is the number of clusters, l_{max} and l_{ave} are the maximum length and the average length of short texts in each corpus.

world, short text corpora to demonstrate the effectiveness of our approach.

4.1. Short text corpora

We used two Chinese short text corpora named ‘Title’ and ‘Q&A’, along with three English short text corpora named ‘Tweet’, ‘Tset’ and ‘SMS’. ‘Title’ and ‘Q&A’ were also used in [14]. ‘Title’ was published by the Sogou Lab[1]. ‘Q&A’ was extracted from a popular Chinese question-and-answer website[2]. ‘Tweet’ and ‘Tset’ were used in [12]. ‘Tweet’ was extracted from the 2011 and 2012 microblog tracks at the Text Retrieval Conference (TREC)[3]. ‘Tset’ was extracted from news titles from a Google news snapshot on November 27, 2013. The clusters of ‘Tweet’ and ‘Tset’ have a query-related construction, while those of ‘Title’ and ‘Q&A’ are topic-related. As such, the semantic concepts of the latter are more general than those of the former. ‘SMS’ is a public set of binary-labeled messages that have been collected for mobile phone spam research[4].

Fig. 1 shows the workflow for preprocessing these short text corpora, where *nd* is the number of short texts containing a given term. For stemming English words, we used the WordNet Lemmatizer from the NLTK package[5]. We found that there were some clusters with only a few (sometimes only one) short texts in ‘Tweet’ and ‘Tset’. Therefore, we extracted one subset from ‘Tweet’, named ‘Tweet_10’, in which each class contained at least 10 short texts. We extracted one subset from ‘Tset’, named ‘T_50’, with at least 50 short texts in each cluster. The basic properties of these short text corpora after processing are shown in Table 1.

4.2. Metrics for evaluating algorithm quality

Since cluster labels are available for the corpora described above, we have used two common external metrics, NMI (Normalized Mutual Information) [11,14,39] and PFM (Pairwise F-Measure, also known as the F1-score) [40], to evaluate the performance of each algorithm.

NMI. Suppose $C = \{C_1, C_2, \ldots, C_k\}$ is a set of k clusters contained in a data set and $C' = \{C'_1, C'_2, \ldots, C'_k\}$ is a set of k clusters obtained by a specific algorithm. NMI is defined as

$$NMI(C, C') = \frac{-2\sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij} log \frac{n \cdot n_{ij}}{n_i^C \cdot n_j^{C'}}}{\sum_{i=1}^{k} n_i^C \log \frac{n_i^C}{n} + \sum_{j=1}^{k} n_j^{C'} \log \frac{n_j^{C'}}{n}}, \tag{16}$$

where n_{ij} is the number of data points in the ground truth cluster C_i that are assigned to the computed cluster C'_j, n_i^C is the number of data points in the ground truth cluster C_i, and $n_j^{C'}$ is the number of data points in the computed cluster C'_j.

PFM. Let T denote the set of data points in the ground truth clusters and W denote the set of clusters assigned by a given algorithm in the corresponding clusters. PFM (Pairwise F-Measure) is defined as follows:

$$PFM = \frac{2 \times precision \times recall}{precision + recall}, \tag{17}$$

where $precision = \|W \bigcap T\| / \|W\|,\ recall = \|W \bigcap T\| / \|T\|$.

4.3. Performance evaluations

We compared our proposed concept decomposition method WordCom with existing state-of-the-art algorithms including heuristic optimization methods TermCut [17], spherical k-means [21] (sp$k$-means)[6], k-means [41] with *tf-idf* weights, Ncut [13], TNMF [14], DNMF [30,31] (dual graph regularized NMF), and probabilistic methods LDA [27][7], BTM [11][8], and GSDMM [12]. All experiments were performed on a personal computer with an Intel 3.20 GHz processor and 16 GB of main memory running Windows 7.0. LDA and BTM were implemented in C++, TermCut and GSDMM were implemented in Java, and WordCom, spherical k-means, k-means, Ncut, TNMF, and DNMF were implemented in Matlab.

As for WordCom, we manually selected 9 initial centers from the decision graphs of ‘Title’, 2 for ‘SMS’, and k initial centers with the top k comprehensive values for the other corpora because there were too many clusters from the decision graphs to be easily counted. We fixed $\alpha = 0.5$ and $\beta = 0.05$ in the experiments because this parameter setting was robust on all corpora in Table 1 (the sensitivity of α and β will be analyzed in the following section). In sp$k$-means, we used the inverse document frequency of words as weights because clustering results were better than those employing *tf-idf* weights on these short text corpora. For the LDA and BTM algorithms, we used the same parameter settings as in [11], where parameters were tuned via grid search for short text corpora (for LDA, $\alpha = 0.05$ and $\beta = 0.01$; for BTM, $\alpha = 50/K$ and $\beta = 0.01$). For GSDMM, we set $\alpha = 0.1$ and $\beta = 0.1$ as was done in [12,42]. We chose 10 for the number of nearest neighbors, used cosine similarity for DNMF, set $\lambda = 0.5$, and set the maximum iteration number at 300.

[1] http://www.sogou.com/labs/.
[2] http://zhidao.baidu.com.
[3] http://trec.nist.gov/data/microblog.html.
[4] http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/.
[5] http://www.nltk.org/.
[6] https://github.com/teammcr192/spherical-k-means.
[7] https://sourceforge.net/projects/gibbslda/.
[8] https://github.com/xiaohuiyan/BTM.

Table 2
The NMI of compared algorithms on real short text corpora.

	Title	Tweet	Tweet_10	Tset	T_50	SMS	Q&A
WordCom	**0.5529**	**0.8766**	**0.8933**	*0.8597*	*0.8581*	*0.5719*	*0.5095*
TermCut	0.3100	0.6694	0.6662	0.6042	0.5942	0.0517	–
sp*k*-means	0.2835 (0.0354)	0.8067 (0.0119)	0.8236 (0.0123)	0.7708 (0.0052)	0.7687 (0.0126)	0.0694 (0.0354)	0.3816 (0.0148)
k-means	0.3613 (0.0412)	0.7801 (0.0081)	0.8056 (0.0116)	0.8035 (0.0044)	0. 8153 (0.0086)	0.1463 (0.0544)	0.4456 (0.0141)
Ncut	0.2321 (0.0197)	0.6247 (0.0131)	0.6517 (0.0060)	0.5355 (0.0093)	0.5485 (0.0064)	0.0245 (0.0039)	0.3565 (0.0045)
TNMF	0.4646 (0.0164)	0.7809 (0.0095)	0.8275 (0.0067)	0.7874 (0.0056)	0.8371 (0.0065)	0.3766 (0.0025)	0.4961 (0.0095)
DNMF	0.4875 (0.0162)	*0.8488* (0.0057)	*0.8860* (0.0048)	0.8201 (0.0039)	0.8400 (0.0092)	0.5617 (0.0007)	0.5047 (0.0109)
LDA	0.4716 (0.0272)	0.7985 (0.0050)	0.8034 (0.0077)	0.8200 (0.0038)	0.8157 (0.0067)	0.5181 (0.1457)	0.4521 (0.0069)
BTM	*0.4926* (0.0144)	0.8041 (0.0087)	0.8184 (0.0098)	**0.8589** (0.0031)	**0.8746** (0.0057)	**0.6548** (0.0067)	**0.5398** (0.0052)
GSDMM	0.0063 (0.0009) 9-9	0.8847 (0.0052) 500-87	0.8804 (0.0063) 100-61	0.8720 (0.0029) 500-123	0.8758 (0.0048) 500-81	0.5567 (0.1957) 2-2	0.5375 (0.0032) 30-30

Table 3
The PFM of compared algorithms on real short text corpora.

	Title	Tweet	Tweet_10	Tset	T_50	SMS	Q&A
WordCom	**0.5921**	**0.7840**	**0.8657**	**0.7628**	**0.7613**	*0.9158*	*0.3715*
TermCut	0.2750	0.2964	0.3664	0.1959	0.2053	0.5872	–
sp*k*-means	0.3150 (0.0474)	0.5923 (0.0525)	0.6848 (0.0455)	0.5283 (0.0178)	0.6089 (0.0267)	0.5925 (0.0137)	0.2342 (0.0171)
k-means	0.3771 (0.0503)	0.5029 (0.0432)	0.6292 (0.0396)	0.5663 (0.0148)	0.6605 (0.0240)	0.6023 (0.0133)	0.3161 (0.0138)
Ncut	0.2938 (0.0266)	0.4684 (0.0248)	0.5521 (0.0118)	0.2263 (0.0170)	0.3485 (0.0152)	0.6891 (0.0245)	0.2247 (0.0074)
TNMF	*0.4930* (0.0166)	0.5547 (0.0400)	*0.7023* (0.0213)	0.5897 (0.0161)	0.7192 (0.0185)	0.7858 (0.0021)	0.3368 (0.0236)
DNMF	0.4562 (0.0130)	*0.6621* (0.0260)	*0.8472* (0.0281)	0.5199 (0.0171)	0.6137 (0.0508)	0.9085 (0.0002)	0.3044 (0.0154)
LDA	0.4513 (0.0291)	0.4747 (0.0168)	0.5498 (0.0268)	0.6002 (0.0111)	0.6674 (0.0168)	0.8755 (0.0743)	0.3324 (0.0103)
BTM	0.4590 (0.0229)	0.4599 (0.0267)	0.5540 (0.0300)	*0.6105* (0.0181)	*0.7279* (0.0203)	**0.9358** (0.0016)	**0.3836** (0.0097)
GSDMM	0.1290 (0.0018)	0.7926 (0.0360)	0.7752 (0.0374)	0.7142 (0.0099)	0.7386 (0.0210)	0.8940 (0.0831)	0.3877 (0.0078)

We used both a graphical method and a confidence interval method [43] to address the issue of replications for the stochastic algorithms sp*k*-means, *k*-means, Ncut, TNMF, DNMF, LDA, BTM and GSDMM. Using the graphical method, we started from 3 replications for each stochastic algorithm, then performed more replications and plotted the cumulative mean of the value of NMI (or PFM) from a series of replications until NMI (or PFM) reached the point at which the line became flat. Simultaneously, we calculated the 95%% confidence interval from these series of replications. These results can be obtained from the WordCom source code package: http://github.com/smileyan448/WordCom.

We found that, for large corpora ('Tset' and 'Q&A'), 10 replications were enough to keep the mean value (of NMI or PFM) of each stochastic algorithm staying in the flat region. Usually, when the variance of the mean value is small, the graph will become a flat line after several replications. For the other corpora, we ran all stochastic algorithms 20 times to keep their NMIs or PFMs in the flat region, with the exception of LDA and GSDMM on 'SMS'. For LDA and GSDMM on the "SMS" corpus, the variances of these two metrics were relatively large and therefore we ran the LDA and GSDMM algorithms 100 times. In Tables 2 and 3, we showed the mean and the standard deviation of NMI and PFM for each stochastic algorithm and the two metrics of the deterministic algorithms WordCom and TermCut on the real corpora, respectively. '–' indicates that TermCut was too slow to process the large-scale corpus 'Q&A'.

It should be noted that GSDMM first set a large cluster number ($K = 500$). The algorithm then converged to a smaller cluster number ($k = 87$). However, the final cluster number detected by GSDMM is not necessarily the same as the ground truth k (see the last line in Table 2, where, for example, the first number of 500-87 is the initial cluster number K, the second number is the average number of output clusters after 10 runs). Therefore, the two metrics of GSDMM are listed as references. Since DNMF does not guarantee finding the cluster structure with the exact number of clusters, it is viewed as a dimensionality reduction method. Thus, the whole process of DNMF for text clustering is followed by a *k*-means step in its original source code. In our experiments, since the accuracy of DNMF without the further step of *k*-means (DNMF-alone) was better than DNMF coupled with *k*-means on 'Title', 'Q&A', and 'SMS', we reported the results of DNMF-alone on these corpora. We marked the best performing algorithm (except for GSDMM) in bold and italicized and underlined the second best algorithm for each corpus in Tables 2–3.

According the three-sigma rule [44], if a data distribution is approximately normal, about 95 percent should be within two standard deviations ($\mu \pm 2\sigma$, where μ is the arithmetic mean and σ is the standard deviation), and about 99.7 percent should lie within

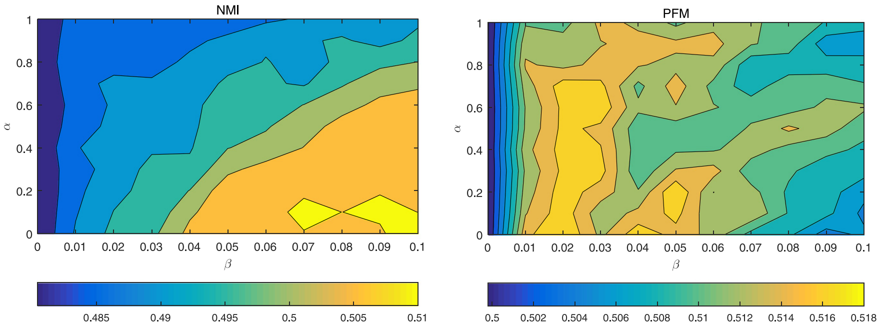


Fig. 2. The sensitivity of α and β on 'Title'.

three standard deviations ($\mu \pm 3\sigma$). As Tables 2 and 3 show, WordCom was always $\mu + 3\sigma$ larger than its improved method spherical k-means on the test corpora. Also, the two metrics of WordCom were larger than the upper intervals of spherical k-means with 95%% confidence on all of the corpora. Thus, WordCom has strong ability to capture the semantic concepts of short text corpora. What is more, it showed the best performance on the corpora 'Title', 'Tweet', 'Tweet_10', 'Tset', and 'T_50' when compared to other algorithms. BTM gave the best performance on 'SMS' and 'Q&A', while WordCom was the second best on these two corpora and all messages in the spam cluster identified by WordCom were correctly clustered. We therefore concluded that WordCom had better performance compared to the other algorithms. BTM had improved LDA on short text clustering and we considered it the second best performing algorithm. TNMF showed good performance in these experiments as well. DNMF (NMF with dual graph regularization) had the ability to maintain the text similarity and term similarity simultaneously, and showed good performance in this study. However, DNMF must be combined with k-means in most cases. This lowers its time efficiency. GSDMM showed good performance, but could not converge at the exact number of clusters.

In addition, WordCom scaled well to large short text corpora such as 'Q&A'. Its running time on 'Q&A' was around 330s, while the concept decomposition method spherical k-means needed around 65,190 s for only one run on this large corpus. The former obtains concept vectors by identifying communities from the word co-occurrence network of short texts. The later extracts concept vectors by finding the centroid of each cluster in document-term vector space. Therefore, it is beneficial to extract concepts from word co-occurrence networks instead of document-term vectors. Moreover, WordCom is a deterministic algorithm, while spherical k-means is sensitive to its initial centers. The source of WordCom and its test short text corpora can be obtained from http://github.com/smileyan448/WordCom as well.

4.4. The sensitivity of α and β

In WordCom we used α and β to balance the contribution of word communities, PageRank centralities of nodes, and community centers. To see the effect of these two parameters on the algorithm, we performed the following experiments to analyze the sensitivity of α and β using grid search.

First, we varied α and β from 0.0 to 1.0 with step sizes 0.1 to test the performance (NMI and PFM) of WordCom on the corpora 'Title', 'Tweet', 'Tset', and 'SMS'. According to our experiments, WordCom is sensitive to β and $\beta \in (0.01, 0.1]$ works well for all of the test corpora. We then let α range from 0 to 1.0 with step sizes of 0.1 and β range from 0 to 0.1 with step sizes of 0.01 to test the performance (NMI and PFM) of WordCom again. The contour graph of the performance (NMI and PFM) at different α and β were shown in Figs. 2–5, where the white region of Figs. 3–4 indicated that the meaningless empty clusters were generated at given α and β.

From Figs. 2–5, we can see that WordCom is not sensitive to α because the concept vectors are mainly derived from community centers $[\mathbf{cen}_{k'}]_{k'=1}^{k}$. We found that the algorithm works very well on all testing corpora when $\alpha = 0.5$ and $\beta = 0.05$. Therefore, we fixed $\alpha = 0.5$ and $\beta = 0.05$ in the algorithm for ease of use, even though we could have obtained better results by fine-tuning α and β.

4.5. Influence of sampling ratio on wordcom_s

As mentioned before, it is possible to extract concept vectors from a subset of large scale short texts, then project original short texts to the concept vectors and obtain the cluster membership indicator for each short text. We tested the influence of the sampling ratio on WordCom_s. For the sake of illustration, we only show the results of 'T_50' and 'Q&A'.

We used the fast uniform random sampling method D [45] to perform sampling. To eliminate the effect of randomness, we ran WordCom_s 10 times at each sampling ratio, which were enough to keep the algorithm staying in the flat region by the graphical method mentioned above. We then plotted the mean and the standard deviation of the two metrics of 'T_50' at the sampling ratio ranging from 10%, 20%, . . . , 100% in Fig. 6a, and plotted those of 'Q&A' at the sampling ratio ranging from 1%, 2%, . . . , 10%, 20%, . . . , 100% in Fig. 6b.

Fig. 6 shows that the accuracy of WordCom_s tends to increase with an increase in the sampling ratio. At the certain sampling ratio, the accuracy of WordCom_s is higher than that of WordCom on the whole corpus. This implies that extracting concept vectors from a subset may allow for a reduction in the noise contained in the original corpus.

Since the reduced processing time of WordCom_s is dependent on the word distribution, we listed the average number of words and running time (time unit is second) for 10 trials at different sampling ratios for 'T_50' and 'Q&A' in Table 4 to demonstrate the effectiveness of WordCom_s. The number of words in the subset of the two corpora proves our assumption that the words in a subset of a large short text corpus may cover most of words in the original corpus. It is clear that the time cost of WordCom is less than that of WordCom_s when the sampling ratio is larger than 60%% on 'T_50'. The reason for this is that WordCom_s requires two extra steps compared to WordCom. Therefore, when the sampling ratio is too large, the time savings on subset will be offset by the computation cost of these extra steps. Therefore, for a short text corpus larger than Q&A, it is necessary to make a trade-off between accuracy and time cost when using WordCom_s.

Using the PAC (probably approximately correct) [46] framework, we can theoretically generate a rough estimation of the small-

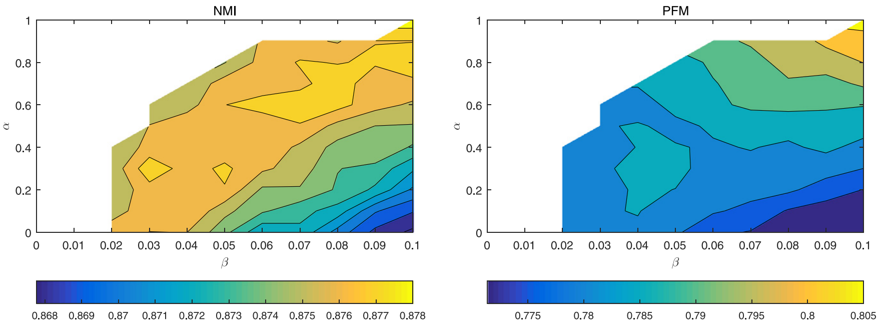


Fig. 3. The sensitivity of α and β on 'Tweet'.

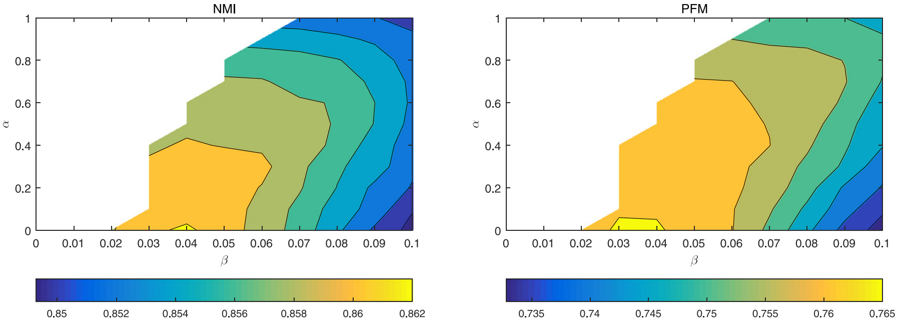


Fig. 4. The sensitivity of α and β on 'Tset'.

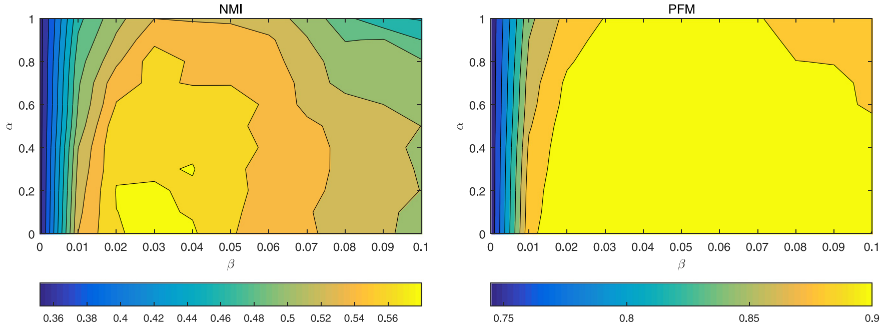


Fig. 5. The sensitivity of α and β on 'SMS'.

Table 4
Word distribution for 'T_50' and 'Q&A' at different sampling ratios.

T_50	10%%	20%%	30%%	40%%	50%%	60%%	70%%	80%%	90%%	100%%
n	919	1838	2757	3676	4595	5514	6433	7352	8271	9190
m	1902	2748	3352	3862	4268	4624	4950	5250	5528	5780
time (s)	19.5	25.0	33.9	48.3	53.9	64.8	84.6	82.8	96.7	61.5
Q&A	1%%	2%%	3%%	4%%	5%%	6%%	7%%	8%%	9%%	10%%
n	935	1870	2806	2741	4677	5612	6547	7482	8414	9357
m	1775	2593	3136	3541	3782	3986	4150	4273	4375	4468
time(s)	112	168	208	238	259	275	290	304	314	327

est sample size n' for a large scale corpus [47] such that $\forall\ \varepsilon \in (0, \frac{1}{2}), \delta \in (0, \frac{1}{2})$,

$$P(\|\bar{X} - \mu\| \leq \varepsilon) > 1 - \delta,$$

where $\bar{X}$ is the mean of a random variable on n' trials and μ is the expectation value of $\bar{X}$ on the whole hypothesis space.

By *Hoeffding Bound*: $\forall\ 0 < \varepsilon < 1 - \mu$,

$$P(\|\bar{X} - \mu\| > \varepsilon) \leq 2e^{-2n'\varepsilon^2},$$

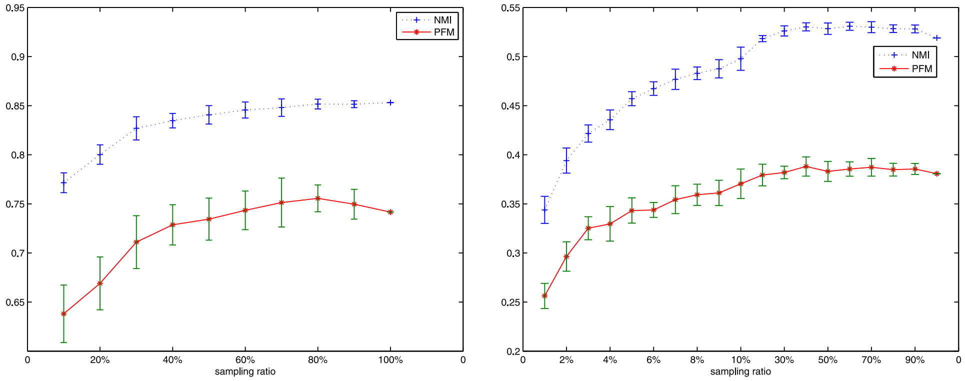
we have

$$n' \geq \frac{1}{2\varepsilon^2} ln \frac{2}{\delta}.$$

Therefore, theoretically, at least 18,444 samples are required to ensure the error rate is at most $\epsilon = 0.01$ with a possibility of $\delta = 95\%$ at minimum. This quantity is close to a sample of "Q&A" at a 20%% sampling ratio. In Fig. 6b, we can see that the accuracy is gurarnteed. This rough estimation is very useful when evaluating millions of short texts. Under this condition, WordCom_s is a good solution.

4.6. Topic visualization

Interpreting topics discovered by a short text clustering method is very important. The topic models LDA and BTM are both capa-



(a) T_50 (b) Q&A

Fig. 6. The influence of the sampling ratio on 'T_50' and 'Q&A'.

Table 5
Topics and their top 10 semantic words identified by WordCom.

Football	Milan	Exposure	Announce	Formal	AC
	champagne	official	lineup	public	early
Science	Discovery	Science	Best	Scientist	Research
	out-space	demystify	nation	picture	photo
Estate	Beijing	2010	House price	Jan.	Real-estate
	Nov.	market	nationwide	city	indemnity
Bonds	German	Treasury bonds	Euro	Italy	Rising
	trading	auction	listing	finance	Spanish
Abroad	China	Study-abroad	World	Student	Education
	immigrant	2011	release	statistics	Hongkong
NECC	NECC	2012	Recruitment	College	University
	independent	enroll	ShangHai	exam	students
Parenting	Baby	Expert	How	Winter	Guide
	issue	health	female	learn	safe
Vogue	Coordinate	Winter-days	EU&US	Star	Fashion
	vogue	keep-warm	street	sweater	warm
Stock	America	Market	Dollar	Economy	Next year
	investment	company	bank	central-bank	Japan

Table 6
Topics and their top 10 semantic words identified by sp*k*-means.

Football	AC	Milan	PY	Announce	Official
	discovery	Earth	record	champagne	FIFA
Science	Spectacular	Sun	Beautiful	Sky	Featured
	out-space	city	control	star	view
Estate					
Bonds	Interest rate	Place	Payment	Tues.	Twenty-five
	element	2010	twenty-four	circulation	2009
Abroad	Student	Twenty-first	Authority	Overseas	Go abroad
	service team	job	event	plenty	Australia
NECC	Offer	Regulation	Select	Art	Excellent
	independent	college	direct admission	recruitment	PKU
Parenting	High level	Athlete	TSU	Kid	Food
	parent	baby	level	topic	drawing
Vogue	Supply	Suffer	Vogue	Recession	Demonstrate
	stock	star	oil price	improvement	warm
Stock	Basis point	Rising	10 years	Earnings	Spanish
	profit rate	Italy	German	success	France
	Cut down	Currency	Active	Next Friday	Decline
	SSM	Moody	expected	increase	profit

PY means the player of the year, PKU stands for Peking University, TSU stands for Tsinghua University, SSM means Shenzhen Stock Market.

ble of learning the distributions of short texts and terms on topics. Concept vectors obtained by spherical *k*-means and WordCom are able to uncover the importance of terms on topics. The matrix factor $U_{m\times k}$ learned by TNMF and DNMF reflects the weights of terms in each of *k* clusters. In this section, we first list the top 10 semantic terms (in the order of their importance) from all 9 clusters found by WordCom and spherical *k*-means on "Title" in Tables 5 and 6, respectively. The topics found by methods LDA, BTM, TNMF, and DNMF were different at each run on account of the influence of their initial values. Therefore, we randomly picked one result each for BTM and TNMF as representative examples since BTM and TNMF performed well (see Tables 2 and 3). The topics and their top 10 semantic terms identified by BTM and TNMF are shown in Tables 7 and 8 (also in the order of terms' importance), respectively. We used 'Title' as an example instead of the other corpora because it only contained 9 clusters with well-labeled topics.

Table 7
Topics and their top 10 semantic words identified by BTM.

Football					
Science	Discovery	Out space	Best	America	Picture
	geography	2011	nation	world	photo
Estate	Beijing	Nov.	Next year	Increase	Season
	house price	America	organization	decrease	global
Bonds	Treasury bonds	Inform	Book-entry	Relate	2011
	related	precautions	payment	interest rate	listing
	Treasury bonds	Dollar	Japan	America	China
	economy	market	buy	central-bank	increase
	Treasury bonds	Profit rate	Auction	Euro	Italy
	Spanish	basis point	2011	German	rising
Abroad	China	Study abroad	America	2011	Beijing
	eduction	immigrant	Dec.	Sohu	nationwide
NECC	Recruitment	2012	University	NECC	Independent
	regulation	enroll	college	TSU	exam
Parenting	Baby	Milan	AC	Expert	How
	kids	winter	NECC	issue	guide
Vogue					
Stock	Trading	Announcement	Bonds	2011	Circulation
	elements	treasury bonds	listing	release	acquisition

Table 8
Topics and their top 10 semantic words identified by TNMF.

Football	AC	Milan	Join	Tycoon	Personal
	drop out	announce	official	MVP	titans
Science	Out space	Picture	Eruption	Volcano	Feature
	Earth	Sun	discovery	giant	spectacular
Estate					
Bonds	Inform	Book-entry	Relate	Related	Precautions
	interest rate	phase II	eighteen	2006	payment
	Profit	Profit rate	Treasury bonds	Auction	Euro
	Italy	subscribe	Spanish	basis point	MP
Abroad	Student	Study abroad	Eduction	22.6%%	H-student
	edu-ministry	Visa	Australia	high school	immigrant
NECC	2012	Two years	Regulation	Recruitment	DA-student
	university	direct admission	traffic	select	talent
Parenting	Baby	Expert	Lecture	How	Health
	food	nursing	Gynecology	health care	nutrition
Vogue	Select	First class	Nov.	2011	Service team
	magnificent	Sohu	nation	Jan.	Beijing
Stock	Next year	Economy	Dollar	Increase	Global
	season	bank	market	investment	central bank

MP stands for multiplying power, edu-ministry stands for education ministry, H-student means high school student, DA-student means directly admitted student.

Since ‘Title’ was a Chinese corpus, we translated all topics and their top 10 Chinese terms in each cluster into English in Tables 5 through 8. In ‘Title’, the nine topics are International Football, Science and Discovery (Science), Real Estate (Estate), Treasury Bonds (Bonds), Study Abroad (Abroad), National College Entrance Examination (NECC), Parenting, Fashion and Vogue (Vogue), and Stock. From Tables 5–8, we can see that WordCom was able to discover all 9 topics contained in ‘Title’, and the semantic terms of each topic agree well with the topic label. Spherical k-means found two clusters of ‘Stock’ and missed one small topic, ‘Estate’. BTM mixed two topics, ‘Football’ and ‘Parenting’, and missed the topic ‘Vogue’ and found three groups of ‘Bonds’ since ‘Bonds’ is the largest cluster in ‘Title’. TNMF mixed the topics ‘Estate’ and ‘Vogue’. Based on this example, we could conclude that WordCom was able to reveal the hidden semantics of each cluster.

5. Conclusions and future considerations

Clustering short text into groups is a challenging task. Further, the semantic concept of each cluster must be revealed. Concept decomposition methods such as spherical k-means use the idea of k-means to group short texts and make use of the centroid of each cluster (named concept vector) to reveal the semantic concept. However, the concept vectors obtained by spherical k-means are extracted from document-term space. When the scale of short text corpus is large, the time cost of spherical k-means is very high. In this study, inspired by the fact that the words in the same concept tend to occur in the same short text and form a densely connected community in the word co-occurrence network, we have presented a new concept decomposition method, WordCom. Concept vectors are extracted from the word co-occurrence network using the community detection method k-rank-D because word communities have the ability to capture complex relationships among words. Although short texts have the problem of high dimensionality, the cardinality of word sets is usually counted within the thousands. This allows the new proposed algorithm WordCom to scale well for large short text corpora. This empirical study has shown that WordCom is more accurate than state-of-the-art algorithms and that the top 10 words based on centrality in each detected word community have revealed the community topic in most cases.

According to our experiments, WordCom is robust to sparse short texts. As the sparseness and shortness of texts increase, WordCom may be a better choice than other algorithms. However, WordCom has difficulty dealing with long texts. This may be caused by the heavily-overlapping community structure of the word co-occurrence network extracted from long texts. Constructing frequently co-occurred word networks rather than word co-occurrence networks may be a solution. This will be explored further in future studies.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (Nos. 61473030 and 61632004). The authors would like to thank the authors of GSDMM and DNMF for sharing their source code and corpora with us. Also, we would like to acknowledge the anonymous reviewers for their constructive comments.

References

[1] S. Banerjee, K. Ramanathan, A. Gupta, Clustering short texts using Wikipedia, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 787–788.

[2] X. Hu, N. Sun, C. Zhang, T.-S. Chua, Exploiting internal and external semantics for the clustering of short texts using world knowledge, in: Proceedings of the ACM Conference on Information and Knowledge Management, ACM, 2009, pp. 919–928.

[3] L. Wang, Y. Jia, W. Han, Instant message clustering based on extended vector space model, LNCS 4683, 2007, 435–443.

[4] M. Sahami, T.D. Heilman, A web-based kernel function for measuring the similarity of short text snippets, in: Proceedings of the International Conference on World Wide Web, ACM, 2006, pp. 377–386.

[5] X.-H. Phan, L.-M. Nguyen, S. Horiguchi, Learning to classify short and sparse text & web with hidden topics from large-scale data collections, in: Proceedings of the International Conference on World Wide Web, ACM, 2008, pp. 91–100.

[6] Y. Song, H. Wang, Z. Wang, H. Li, W. Chen, Short text conceptualization using a probabilistic knowledgebase, in: Proceedings of the International Joint Conference on Artificial Intelligence, AAAI Press, 2011, pp. 2330–2336.

[7] O. Jin, N.N. Liu, K. Zhao, Y. Yu, Q. Yang, Transferring topical knowledge from auxiliary long texts for short text clustering, in: Proceedings of the ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 775–784.

[8] J. Tang, X. Wang, H. Gao, X. Hu, H. Liu, Enriching short text representation in microblog for clustering, Front. Comput. Sci. 6 (1) (2012) 88–101.

[9] A. Hindle, J. Shao, D. Lin, J. Lu, R. Zhang, Clustering web video search results based on integration of multiple features, in: Proceedings of International Conference on World Wild Web, 2011, pp. 53–73.

[10] D. Milne, O. Medelyan, I.H. Witten, mining domain-specific thesauri from wikipedia: a case study, in: Proceedings of ACM International Conference on Web Intelligence, Hongkong, 2006, pp. 442–448.

[11] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2013, pp. 1445–1456.

[12] J. Yin, J. Wang, A Dirichlet multinomial mixture model-based approach for short text clustering, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 233–242.

[13] X. Yan, J. Guo, S. Liu, X.-q. Cheng, Y. Wang, Clustering short text using Ncut-weighted non-negative matrix factorization, in: Proceedings of the ACM international Conference on Information and Knowledge Management, ACM, 2012, pp. 2259–2262.

[14] X. Yan, J. Guo, S. Liu, X. Cheng, Y. Wang, Learning topics in short texts by non-negative matrix factorization on term correlation matrix, in: Proceedings of the SIAM International Conference on Data Mining, SIAM, 2013.

[15] S. Seifzadeh, A.K. Farahat, M.S. Kamel, F. Karray, Short-text clustering using statistical semantics, in: Proceedings of the International Conference on World Wide Web, 2015, pp. 805–810.

[16] X. Huang, Y. Ye, X. Du, S. Deng, Short text clustering with expanding keywords through concept graph, J. Comput. Inf. Syst. 9 (21) (2013) 8649–8657.

[17] X. Ni, X. Quan, Z. Lu, L. Wenyin, B. Hua, Short text clustering by finding core terms, Knowl. Inf. Syst. 27 (3) (2011) 345–365.

[18] S. Osiński, D. Weiss, A concept-driven algorithm for clustering search results, Intell. Syst. IEEE 20 (3) (2005) 48–54.

[19] O. Zamir, O. Etzioni, Web document clustering: a feasibility demonstration, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1998, pp. 46–54.

[20] G. Murphy, The Big Book of Concepts, MIT press, 2004.

[21] I.S. Dhillon, D.S. Modha, Concept decompositions for large sparse text data using clustering, Mach. Learn. 42 (1–2) (2001) 143–175.

[22] A. Veling, P. Van Der Weerd, Conceptual grouping in word co-occurrence networks, in: Proceedings of the International Joint Conference on Artificial Intelligence, 99, 1999, pp. 694–701.

[23] D. Jurgens, Word sense induction by community detection, in: Proceedings of Graph-based Methods for Natural Language Processing, ACL, 2011, pp. 24–28.

[24] Y. Li, C. Jia, J. Yu, A parameter-free community detection method based on centrality and dispersion of nodes in complex networks, Phys. A 438 (2015) 321–334.

[25] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[26] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1999, pp. 50–57.

[27] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[28] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Informaion Retrieval, ACM, 2003, pp. 267–273.

[29] W. Xu, Y. Gong, Document clustering by concept factorization, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2004, pp. 202–209.

[30] F. Shang, L. Jiao, F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, Pattern Recognit. 45 (6) (2012) 2237–2250.

[31] J. Ye, Z. Jin, Dual-graph regularized concept factorization for clustering, Neurocomputing 138 (2014) 120–130.

[32] L. Bottou, Y. Bengio, Convergence properties of the k-means algorithms, in: Advances in Neural Information Processing Systems, 1995, pp. 585–592.

[33] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.

[34] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.

[35] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web, Stanford Digital Libraries Working Paper (1998) 1–17.

[36] Y. Hu, P. Zhang, Y. Fan, Z. Di, Community detection by signaling on complex networks, Phys. Rev. E 78 (1) (2008) 016115.

[37] Y. Jiang, C. Jia, J. Yu, An efficient community detection method based on rank centrality, Phys. A-Stat. Mech. Appl. 392 (9) (2013) 2182–2194.

[38] E.A. Leicht, P. Holme, M.E. Newman, Vertex similarity in networks, Phys. Rev. E 73 (2) (2006) 026120.

[39] A. Fred, A. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans Pattern Anal. Mach Intell. 27 (6) (2005) 835–850.

[40] B. Larsen, C. Aone, Fast and effective text mining using linear-time document clustering, in: Proceedings of KDD, 1999, pp. 16–22.

[41] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recogn. Lett. 31 (8) (2010) 651–666.

[42] J. Yin, J. Wang, A text clustering algorithm using an online clustering scheme for initialization., in: Proceedings of KDD, 2016, pp. 1995–2004.

[43] S. Robinson, Simulation: The Practice of Model Development and Use, Palgrave Macmillan, 2014.

[44] F. Pukelsheim, The three sigma rule, Am. Stat. 48 (2) (2012) 88–91.

[45] J.S. Vitter, An efficient algorithm for sequential random sampling, ACM Trans. Math. Softw. 13 (1) (1987) 58–67.

[46] L. Valiant, Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World, Basic Books, 2013.

[47] W. Hoeffding, Lower bounds for the expected sample size and the average risk of a sequential procedure, Ann. Math. Stat. 31 (2) (1994) 359–375.

Ciayan Jia received her Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, PR China, in July 2004. She had been a postdoctoral fellow in the Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai, PR China, in 2004–2007. She is now a professor in the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, PR China. Her current research interests include social computing, text clustering, community detection in networks, etc.