Information Sciences 260 (2014) 37–50

Contents lists available at ScienceDirect

Information Sciences

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

Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization

Libin Yang [a], Xiaoyan Cai [a,*], Yang Zhang [a], Peng Shi [b,c]

[a] College of Information Engineering, Northwest Agriculture and Forestry Univerisity, Xi'an Shaanxi, China
[b] College of Engineering and Science, Victoria University, Melbourne, VIC 8001, Australia
[c] Department of Computing and Mathematical Sciences, University of South Wales, Pontypridd CF37 1DL, United Kingdom

ARTICLE INFO

Article history:
Received 24 July 2013
Received in revised form 19 October 2013
Accepted 19 November 2013
Available online 28 November 2013

Keywords:
Ranking-based clustering
Sentence clustering
Theme-based summarization

ABSTRACT

Sentence clustering plays a pivotal role in theme-based summarization, which discovers topic themes defined as the clusters of highly related sentences in order to avoid redundancy and cover more diverse information. As the length of sentences is short and the content it contains is limited, the bag-of-words cosine similarity traditionally used for document clustering is no longer reasonably suitable. Special treatment for measuring sentence similarity is necessary. In this paper, we propose a ranking-based clustering framework that utilizes ranking distribution of documents and terms to help generate high quality sentence clusters. The effectiveness of the proposed framework is demonstrated by both the cluster quality analysis and the summarization evaluation conducted on the DUC 2004 and DUC2007 datasets.

1. Introduction

With the rapidly growing popularity of the Internet and a variety of information services, obtaining the desired information within a short amount of time becomes a serious problem in the information age. As such, new technologies that can process information efficiently are in great need. Automatic document summarization, which is a process of reducing the size of documents while preserving the important semantic content, is an essential technology to overcome this obstacle [1,2]. Most of the summarization work done till date is based on the sentence extraction framework, which ranks sentences according to various pre-specified criteria and then selects the most salient sentences from the original documents to form a concise summary.

In addition to sentence salience, the other two fundamental issues that should be addressed in summarization are information redundancy and information diversity, two sides of the same coin [3]. When all the given documents are supposed to be about the same topic, they are very likely to repeat some important information in different documents or different places in a document. Therefore, effectively recognizing sentences with the same or very similar content is necessary for reducing redundancy and covering more diverse informative content in a summary. This is normally achieved by clustering highly related sentences into topic themes. Summaries can then be produced, for example, by extracting representative sentence(s) from each theme cluster. Thus, good sentence clusters are the guarantee of good summaries in theme-based summarization.

The key part in sentence clustering is to estimate the similarity between two sentences [4]. Intuitively, many similarity measures traditionally used for document clustering cannot be directly applied to sentence clustering. The solutions that rely on term overlaps can be effective when dealing with documents because the documents about the same topic may share

* Corresponding author. Tel.: +86 29 8707 4975.
E-mail addresses: libiny@nwsuaf.edu.cn (L. Yang), xiaoyanc@nwsuaf.edu.cn (X. Cai), zhangyang@nwsuaf.edu.cn (Y. Zhang), peng.shi@vu.edu.au (P. Shi).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.11.026

many terms in common. However, the sentences with very similar meanings do not necessarily share enough terms, owing to the short length and the limited content that they contain. Inevitably, the similarity measures based on term overlaps alone fail to perform well in sentence clustering.

To help alleviate this problem, we argue in this paper that a term can be deemed as an independent text object instead of a feature of a sentence. Based on it, a ranking-based sentence clustering framework is developed. In the framework, conditional ranks of documents and terms help to get generative probability of each sentence, so sentences can be mapped into a very low dimensional space defined by current clustering result. Then each sentence in these clusters will be readjusted based on the new measure. During each iteration, clustering results will be improved under new measure space. The experimental results show that the framework is able to generate more reasonable sentence clusters that, in turn, lead to more meaningful summarization performance.

The three main contributions of the paper are:

(1) Two different ranking functions are defined in a document tri-type star graph constructed from the given document set, namely simple ranking and authority ranking, respectively. Based on these two ranking functions, conditional ranks of documents and terms can be calculated, which are served as features for generative probability of each sentence, so sentences can be mapped into a very low dimensional space.
(2) Ranking-based sentence clustering framework is developed. The framework avoids defining and calculating pairwise similarity between sentences, or between terms or between documents, but maps each sentence into a very low dimensional space defined by current clustering result. Then each sentence in these clusters will be readjusted based on the new measure. During each iteration, clustering results will be improved under new measure space and the quality of measure will be improved since it is derived from better clusters.
(3) Thorough experimental studies using intrinsic cluster quality evaluation method and extrinsic summarization method are conducted to verify the effectiveness and robustness of the proposed approach.

The rest of this paper is organized as follows. Section 2 reviews related work on sentence clustering for summarization. Section 3 defines document tri-type star graph and two ranking functions. Section 4 presents ranking-based sentence clustering framework and their application to multi-document summarization. Section 5 addresses experiments and evaluations. Conclusions are presented in Section 6.

2. Related work on sentence clustering for summarization

All clustering algorithms require measuring similarity/dissimilarity between the objects to be clustered. Measuring similarity between two sentences has been previously studied in both Information Retrieval (IR) and Natural Language Processing (NLP) communities, with the emphasis on sentence representation. This section reviews some related work in order to explore the strengths and limitations of previous methods, and to identify the particular difficulties in computing sentence similarity. Related work can roughly be classified into three major categories: term co-occurrence method, corpus-based method and descriptive features-based method.

Traditional term co-occurrence method constitutes a vector space corresponding to all the content terms observed in the documents (stop terms are excluded and remaining content terms are stemmed) and then constructs a sentence-by-term matrix, elements of the matrix denotes the weight of a term in a sentence [5–7]. The term weight can be estimated using TFISF as described in [8]. Given a sentence-by-term matrix, similarity between two sentences can be measured by the cosine matches based on surface term matching, which are derived from the corresponding row vectors in the matrix. Once the sentence similarity matrix is obtained, any classical clustering algorithms can be performed on it. Besides individual terms, other term compounds, such as name entities [9], bigrams and trigrams [10] are also extracted from the original text to construct the feature space.

Unlike full texts of documents, short text snippets, such as sentences, commonly contain no more than fifty terms. Because of the short length, the bag-of-words representation of individual sentence is likely to be very sparse, and thus it cannot provide enough contexts to obtain a reliable similarity measure. This makes conventional text clustering approaches fail to achieve satisfactory results on sentence clustering and presents a great challenge to us. A good representation of sentence is expected be able to capture the embedded semantic information and the shared contextual information of it. Thus a solution to overcome the weakness of term co-occurrence method is to explore the enriched sentence representations, such as concept- or context- enriched representations. These methods are fall into corpus-based category.

Zhao et al. [11] used WordNet as a semantic resource to build a semantic-based vector instead of a term-based one for representing a sentence. For each content term in a sentence, its correlative concepts (e.g., synonym, hypernym, homonym, etc.) in WordNet are extracted and added into the original sentence vector as concept entries. Similarity of sentences is measured by the cosine similarity of their corresponding row vectors in the new expanded matrix. In this way, semantic information that is not explicitly expressed in the sentences can be captured. Therefore, sentences with different but semantically related meaning can be related together. Along the same line of thought, Banerjee et al. [12] proposed to improve the accuracy of short texts clustering by enriching their representation with additional features derived from Wikipedia. They first downloaded the English Wikipedia dump of one day, and then removed templates, articles describing Wikipedia features

Table 1
Current work in sentence similarity.

Methods	Refs.
Word co-occurrence methods	[5–10]
Corpus-based methods	[11–15,22,16–19]
Descriptive feature-based methods	[34–36]

and articles containing less than 50 non-stop words. After that, they created a Lucene index of these Wikipedia articles and used each sentence as a query to retrieve the top 10 matching Wikipedia articles from the Lucene index. The titles of the retrieved Wikipedia articles which are referred as Wikipedia concepts now serve as additional features of the sentence. The list of 10 Wikipedia concepts was then represented by a vector where the weight of a concept is the frequency of the concept in the list. A new representation of the sentence was generated by augmenting this vector to the surface word representation. However, the approach using Wikipedia concept heavily relies on the effectiveness of search engines and most important it is time-consuming.

Latent Semantic Analysis (LSA) [13] is a fully automatic mathematical technique for extracting and representing the contextual usage of terms' meaning in passages of discourse. It has been applied to sentence clustering by Islam and Inkpen [14]. The basic idea underlying LSA-based clustering is that the aggregate of all the term contexts in which a given term does or does not appear provides mutual constraints that largely determine the similarity of meaning of terms and sets of terms to each other. Based on a sentence-by-term matrix, LSA-based sentence clustering applies Singular Value Decomposition (SVD) to transform the sentence-by-term matrix, into a sentence-by-concept vector matrix, a singular values matrix, and a concept-by-term vector matrix. SVD can capture the interrelationships among terms so that terms and sentences can be clustered on a "semantic" basis rather than on the basis of terms only. Similarity of sentences is then measured by the cosine of the angle between the corresponding reduced row vectors [15].

Unlike LSA, Hyperspace Analogues to Language (HAL) [33] builds a term-by-term matrix based on word co-occurrences within a moving window of a predefined width. The window (typically with a width of 10 terms) moves over the entire text of the corpus. An $N \times N$ matrix is formed for a given vocabulary of N terms. Each entry of the matrix records the (weighted) term co-occurrences within the window moving through the entire corpus. The meaning of a word is then represented as a $2N$-dimensional vector by combining the corresponding row and column in the matrix. Subsequently, a sentence vector is formed by adding together the word vectors for all words in the sentence. Similarity between two sentences is calculated using a metric such as Euclidean distance.

While the previous enrichments attempt to transform the term-based sentence vector into the concept-based sentence vector, some researchers considered not only sentence information but also paragraph information which contains more contextual information [16–18]. Cai and Li [16] involved sentence context into the sentence representation. That is, in addition to the sentence itself, one preceding sentence and one following sentence are also included and a new sentence-by-term matrix is constructed. Sentence cosine similarity is then calculated using this new matrix. Besides, Cai and Li [19] deemed that documents and terms both can be used as independent text objects. Sentence clustering performance will be improved if the local semantic interpretation carried by terms and the global background information derived from documents can be better utilized. Based on these assumptions, they developed integrated and interactive co-clustering frameworks to cluster sentences, terms and documents together such that the performance of sentence clustering, term clustering and document clustering are mutually enforced.

The third category of related work is the descriptive features-based methods. The feature vector method tries to represent a sentence using a set of predefined features [34]. Basically, a term in a sentence is represented using semantic features, such as, nouns may have features such as HUMAN (with value of human or nonhuman), SOFTNESS (soft or hard), and POINTNESS (pointed or rounded). A variation of feature vector method is the introduction of primary features and composite features [35,36]. Primary features are those primitive features that compare single items from each text unit. Composite features are the combination of pairs of primitive features. A text is then represented in a vector consisting of values of primary features and composite features.

Table 1 shows current works in sentence similarity and also the technique used by each work.

3. Problem formulation

3.1. Document tri-type star graph

In this section, we first present the document-sentence-term tri-type star graph model for a set of given documents D, based on which the ranking-based sentence clustering framework is developed. Let $G = \langle V, E, W \rangle$, where V is the set of vertices that consists of the document set $D = \{d_1, d_2, \ldots, d_b\}$, sentence set $S = \{s_1, s_2, \ldots, s_n\}$ and the term set $T = \{t_1, t_2, \ldots, t_m\}$, i.e., $V = D \cup S \cup T$, b is the number of documents, n is the number of sentences and m is the number of terms. Each term vertex is the sentence that is given in the WordNet as the description of the term, it is extracted the first sense used from WordNet

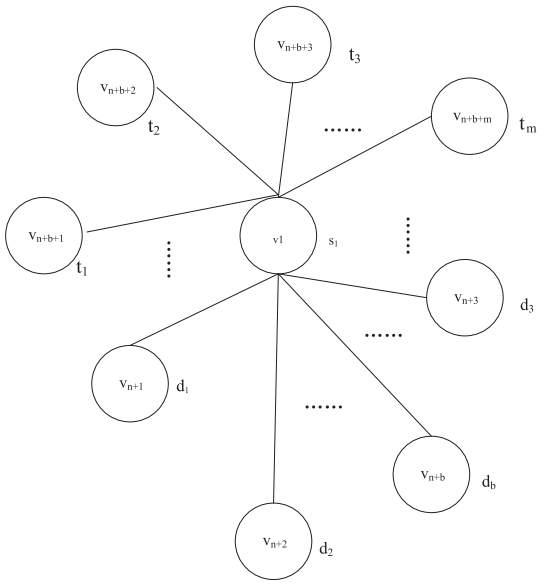

Fig. 1. Illustration of document tri-type star graph.

instead of the word itself.[1] E is the set of edges that connect the vertices, an edge can connect a sentence to a term, or a sentence to a document, i.e., $E = \{\langle v_i, v_j \rangle | v_i \in S, v_j \in D \cup T\}$. The graph G is presented in Fig. 1. For ease of illustration, we only demonstrate the edges between a sentence vertex v_1 and other vertices which represent documents or terms. W is the adjacency matrix which is formulated as

$$W = [w_{ij}]_{n\times(m+b)} \tag{1}$$

where the element w_{ij} represents the weight of the edge connecting v_i and v_j, which is defined as the cosine similarity between the sentence s_i and the document d_j (or term t_j).

In graph G, S is called the center type, D and T are called attribute types. Attribute types only have links to the center type. That's why the graph G is called tri-type star graph. During clustering, sentences are the objects first to be clustered at each iteration, and links to the documents and terms are used to help clustering sentence.

Definition 1. Theme cluster. Given a graph G, a theme cluster C is defined as $C = \langle G', p_T \rangle$, where G' is a sub-graph of G, i.e., $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$ and $W(G') = W(G)$. Function $p_C: V(G') \to [0, 1]$ is defined on $V(G')$, $0 \leqslant p_C(v) \leqslant 1$, which denotes the probability that v belongs to cluster C, i.e., $P(v \in C)$.

For simplicity, we use $V(C)$ to denote the vertices set $V(G')$ in graph G' and $E(C)$ to denote the edge set $E(G')$. Also, for $v \notin V(G')$, we define $p_C(v) = 0$.Based on the definition of theme cluster, a theme cluster is a sub-graph integrating statistical information for vertices. We apply the idea of soft clustering in a theme cluster, which means for each vertex $v \in V(C)$, it can belong to several clusters with some probability $p_{C_k}(v)$ $(k = 1, \ldots, K)$ and

$$\sum_{k=1}^{K} p_{C_k}(v) = 1 \tag{2}$$

where K is the number of theme clusters. But for target objects v, we restrict $p_C(v)$ as either 0 or 1, and they can belong to merely one theme cluster. For each theme cluster, we argue that it has much simpler structure and can be modeled as a ranking-based generative model. Therefore, every theme cluster is corresponding to a generative model, according to which generative probabilities of every target object in each cluster can be calculated.

3.2. Basic ranking functions of attribute types

Recall that our goal is to obtain more accurate sentence clusters and generate good summaries in theme-based summarization. In this paper, conditional ranks of documents and terms are served as features for generative probability of each sentence, so sentences can be mapped into a very low dimensional space. Then each sentence will be readjusted based on the new space which can further improve clustering performance. In this section, we propose two ranking functions.

[1] For example, if the term is 'cat', then the corresponding term vertex in the graph should be 'feline mammal usually having thick soft fur and being unable to roar; domestic cats; wildcats' extracted from the WordNet.

(1) Simple Ranking

In a given network G, let $p(v|O_v, G)$ denote the simple ranking distribution of a document/term in its own type, in which v is an object from type O_v,$O_v \in D \cup T$. $p(v|O_v, G)$ is calculated as the simple occurrence counting for each object normalized in its own type, i.e.

$$p(v|O_v, G) = \frac{\sum_{x \in N_G(v)} W_{vx}}{\sum_{x' \in O_v} \sum_{x \in N_G(v')} W_{v'x}} \tag{3}$$

where $N_G(v)$ is the neighborhood of object v in the graph G and $N_G(v) \subseteq S$.

In our document tri-type star graph, the rank score for a document using simple ranking will be proportional to the sum of ranking scores of its contained sentences.

Property 1. *Given a document tri-type star graph $G = \langle D \cup S \cup T, E, W\rangle$, where S is the center type, and $\forall s, N_G(s) = \{d, t\}(d \in D, t \in T)$, the expected coding error for estimating the joint probability of $P(D, T)$ by generative model for G under simple ranking $P(D)$ and $P(T)$ is $I(D, T)$, where $I(D, T)$ is the mutual information between D and T.*

Proof.

$$\begin{aligned} \varepsilon &= \sum_{d \in D} \sum_{t \in T} p(d,t)[\log p(d,t) - \log \hat{p}(d,t)] \\ &= \sum_{d \in D} \sum_{t \in T} p(d,t)[\log p(d,t) - \log p(d)p(t)] = I(D,T) \end{aligned} \tag{4}$$

□

The above equation illustrates that, if a type of attribute objects has a small mutual information with other types of attribute objects, simple ranking is good for it.

(2) Authority ranking

Authority ranking function considers the authority propagation of vertices in the graph. For the document tri-type star graph G, the propagation of authority score from Type D to Type T through the center type S is defined as

$$P(T|O_T, G) = W_{TS} W_{SD} P(D|O_D, G) \tag{5}$$

where W_{TS} and W_{SD} are the weight matrices between terms and sentences, sentences and documents respectively. Generally, authority score of one type of objects could be a combination of scores from different types of objects [20]. It turns out that the iteration method of calculating ranking distribution is the power method to calculate the primary eigenvector of a square matrix denoting the strength between pairs of objects in that certain type, which can be achieved by selecting a walking path (or a combination of multiple paths) in the graph.

Property 2. *Given a document tri-type star graph $G = \langle D \cup S \cup T, E, W\rangle$, where S is the center type, and $\forall s, N_G(s) = \{d, t\}(d \in D, t \in T)$, authority ranking $P(D)$ and $P(T)$ are calculated through* Eq. (3) *iteratively, then estimated joint distribution $\widehat{P}(D,T) = \{\hat{p}(d,t) = P(D=d)P(T=t), d \in D, t \in T\}$ equals to the joint distribution represented by one rank matrix $\frac{M}{||M||_1}$, such that $||W_{DS}W_{ST} - M||_F$ is minimized.*

Proof. Let $USV^T = W_{DS}W_{ST}$ be SVD of $W_{DS}W_{ST}$, and U_1 and V_1 be the first columns of U and V corresponding to the largest singular value σ_1, according to Eckart-Young theorem, $M = \sigma_1 U_1 V_1^T = \min_{\widetilde{M}} ||W_{DS}W_{ST} - \widetilde{M}||$, where $rank(\widetilde{M}) = 1$. According to the authority ranking, $P(D) = U_1/||U_1||_1$ and $P(T) = V_1/||V_1||_1$, thus $M/||M||_1 = \frac{\sigma_1 U_1 V_1^T}{||\sigma_1 U_1 V_1^T||_1} = P(D)P(T)^T$, where $||M||_1$ is entry-wise 1-norm of M. □

From the property 2, we can obtain an intuition that the authority ranking is able to catch the largest component structure of a graph under the constraints that the relations between objects are recovered by 1-dimensional ranking. In the given document set, according to the rules that (1) highly ranked documents contain many highly ranked sentences which contain many highly ranked terms, and (2) highly ranked terms appear in highly ranked sentences which also appear in highly ranked documents, we determine the iteration equation as:

$$\begin{aligned} P(D|O_D, G) &= W_{DS} D_{ST}^{-1} W_{ST} P(T|O_{T,G}) \\ P(T|O_T, G) &= W_{TS} D_{SD}^{-1} W_{SD} P(D|O_D, G) \end{aligned} \tag{6}$$

where D_{ST} and D_{SD} are the diagonal matrices with the diagonal value equaling to row sum of W_{ST} and W_{SD}. Since all these matrices are sparse, in practice, the rank scores of objects need only be calculated iteratively according to their limited neighbors.

Now we have document and term ranks over the graph G and are ready to introduce the ranking-based sentence clustering framework.

4. Ranking-based sentence clustering framework

In this section, we introduce a ranking-based sentence clustering framework. The major difficulty that lies in clustering in tri-type star graph is the definition and calculation of similarity between each pair of objects. The general idea of the framework is to avoid defining and calculating pairwise similarity between sentences, or between terms, or between documents, but map each sentence into a very low dimensional space defined by current clustering result. Then each sentence in these clusters will be readjusted based on the new measure. During each iteration, clustering results will be improved under new measure space and the quality of measure will be improved since it is derived from better clusters. The framework can be illustrated in Fig. 2 as follow:

Based on the Fig. 3, we develop ranking-based probabilistic for each theme cluster in Section 4.1, and present the posterior probabilities for each sentence in Section 4.2. The above two processes are repeated until sentence clusters do not change significantly. Then we can get the final sentence clusters which are presented in Section 4.3.

4.1. Ranking-based probabilities generative model for each theme cluster

Sun et al. [21] analyzed DBLP dataset and found that 7.64% of the most productive authors publish 74.2% of all the papers, among which 56.72% papers are published in merely 8.62% of the biggest venues, which means large size conferences and productive authors are intended to co-appear via papers. We extend the heuristic by using ranking, which denotes the overall importance of an object in a graph, instead of degree. The intuition is that degree may not represent global importance of an object well. Under this observation, we simplify the graph structure by proposing a probabilistic generative model for sentences, where a set of highly ranked documents and terms are more likely to co-appear to generate a sentence.

We factorize the impact of documents and terms and then model the generative behavior of sentences. The idea of factorizing a graph is: we assume that given a graph G, the probability to visit sentences from documents and terms are

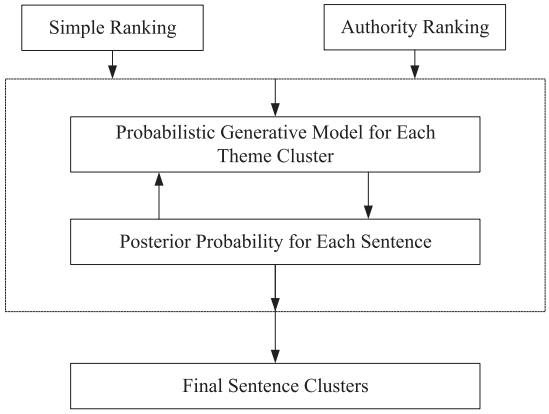

Fig. 2. Ranking-based sentence clustering framework.

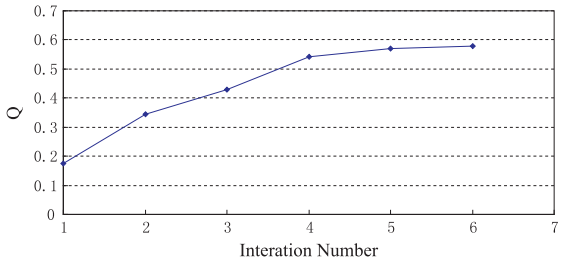

Fig. 3. Trends of the cluster quality with increased iteration numbers on the DUC2004 dataset.

independent to each other. Still, the probability to visit a term (or a document) in G, say term t_i (or document d_i), $p(t_i|G)$ (or $p(d_i|G)$) can be decomposed into two parts: $p(t_i|G) = p(T|G) \times p(t_i|T, G)$ (or $p(d_i|G) = p(D|G) \times p(d_i|D, G)$), where the first part $p(T|G)$ (or $p(D|G)$) is the overall probability that type of term (or document) will be visited in G, and the second part $p(t_i|T, G)$ (or $p(d_i|D, G)$) is the probability that t_i (or d_i) will be visited among all terms (or documents) in the graph G. Generally, given an attribute object v and its type O_v ($O \in D \cup T$), the probability to visit v in G is defined as in Eq. (7):

$$p(v|G) = p(O_v|G) \times p(v|O_v, G) \tag{7}$$

In practice, $p(O_v|G)$ can be estimated by the proportion of objects in O_v compared with the whole attribute object set $D \cup T$ for all terms and documents. We will show that the value of $p(O_v|G)$ is not important and can be set to 1 in the following sub-section. Ranking distribution $p(v|O_v, G)$ for attribute type O_v in a given graph G has been addressed in Section 3.2.

Also, we make another independence assumption that within the same type of objects, the probability to visit two different objects is independent to each other.

$$p(v_i, v_j|O_v, G) = p(v_i|O_v, G) \times p(v_j|O_v, G) \tag{8}$$

where $v_i, v_j \in O_v$ and O_v is either term type or document type.

Now, we build the generative model for sentences given the ranking distributions of documents and terms in the graph G. A sentence s_i is determined by several attribute objects, say $v_{i1}, v_{i2}, \cdots, v_{in_i}$, where n_i is the number of links s_i has. The probability to generate a sentence s_i is equivalent to generating these attribute objects with the occurrence number indicated by the weight of the edge. Under the independency assumptions that we have made, the probability to generate a sentence s_i in the graph G is defined as follows:

$$p(s_i|G) = \prod_{v \in N_G(s_i)} p(v|G)^{w_{s_i,v}} = \prod_{v \in N_G(s_i)} p(v|O_v, G)^{w_{s_i,v}} p(O_v|G)^{w_{s_i,v}} \tag{9}$$

where $N_G(s_i)$ is the neighborhood of sentence s_i in graph G, and O_v is used to denote the type of object v, it is either term type or document type. Intuitively, a sentence is generated in a cluster with high probability, if the document it is contained and terms appeared in the sentence all have high probability in that cluster.

4.2. Posterior probability for sentences

Once we get the generative model for each theme cluster, we can calculate posterior probabilities for each sentence. Now the problem becomes that suppose we know the generative probabilities for each sentence generated from each theme cluster C_k, $k = 1, 2, \ldots, K$, K is the cluster number given by user. We will calculate K posterior probabilities for each sentence. The generative model for sentences in G plays a role as background model. In this sub-section, we will introduce the method to calculate posterior probabilities for sentences.

According to the generative model for sentences, the generative probability for a sentence s_i in the target type S in a sub-graph $G_k = G(C_k)$ can be calculated according to the conditional rankings of attribute types in that sub-network:

$$p(s_i|G_k) = \prod_{v \in N_{G_k}(s)} p(v|O_v, G_k)^{w_{s_i,v}} \cdot p(Y_v|O_k)^{w_{s_i,v}} \tag{10}$$

where $N_{G_k}(s_i)$ denotes for the neighborhood of object s_i in sub-graph G_k and $N_{G_k}(s_i) \in D \cup T$. In Eq. (10), in order to avoid zero probabilities [22] in conditional rankings, each conditional ranking should be smoothed using global ranking with smoothing parameter λ, before calculating posterior probabilities for sentences:

$$P(V|O_V, G_k) = (1 - \lambda)P(V|O_V, G_k) + \lambda P(V|O_V, G) \tag{11}$$

where λ is a parameter that denotes how much we should utilize the ranking distribution from global ranking.

Once a clustering is given on the original graph G, i.e. $C_1, C_2, \ldots, C_K$, we can calculate the probability for each sentence simply by Bayesian rule:

$$p(C_k|s_i) \propto p(s_i|C_k) \times p(C_k) \tag{12}$$

where $p(s_i|C_k)$ is the probability that sentence s_i generated from theme cluster C_k, and $p(C_k)$ denotes the relative size of cluster C_k, i.e., the probability that a sentence belongs to cluster C_k ($k = 1, 2, \ldots, K$) overall. From this formula, we can see that type probability $p(O_v|G)$ is just a constant for calculating posterior probabilities for sentences and can be neglected.

In order to get the potential cluster size $p(C_k)$ for each theme cluster C_k, we choose cluster size $p(C_k)$ that maximizes log-likelihood to generate the whole collection of sentences and then use EM algorithm to get the local optimum for $p(C_k)$.

$$\log L = \sum_{i=1}^{n} \log(p(s_i)) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} p(s_i|C_k)p(C_k) \right] \tag{13}$$

We use the EM algorithm to get $p(C_k)$ by simply using the following two iterative formulas:

$$p^{(t)}(C_k|s_i) \propto p(s_i|C_k)p^{(t)}(C_k); \quad p^{(t+1)}(C_k) = \sum_{i=1}^{n} p^{(t)}(C_k|s_i)/n \tag{14}$$

Initially, we can set $p^{(0)}(C_k) = \frac{1}{K}$.

When posterior probability is calculated for each sentence in each Theme Cluster, each sentence s_i can be represented as a K dimensional vector: $\vec{s}_i = (p(C_1|s_i), p(C_2|s_i), \ldots, p(C_k|s_i))$.

4.3. Similarity measure

As sentence s_i is represented as a K dimensional vector $\vec{s}_i = (p(C_1|s_i), p(C_2|s_i), \cdots, p(C_k|s_i))$. The center of each theme cluster can thus be calculated accordingly, which is the mean of $\vec{s}_i$ for all s_i in the same cluster, i.e.,

$$\overrightarrow{\mathbf{Center}}_{C_k} = \frac{\sum_{s_i \in C_k} \vec{s}_i}{|C_k|} \tag{15}$$

where $|C_k|$ is the size of C_k.

Then the similarity between each sentence and each cluster can be calculated as the cosine similarity between them, i.e.,

$$sim(s_i, C_k) = \frac{\langle \vec{\mathbf{s}}_\mathbf{i}, \overrightarrow{\mathbf{Center}}_{\mathbf{C_k}} \rangle}{||\vec{\mathbf{s}}_\mathbf{i}||^2 \cdot ||\overrightarrow{\mathbf{Center}}_{\mathbf{C_k}}||^2}. \tag{16}$$

Finally, each sentence is re-assigned to a cluster that is the most similar to the sentence. Based on the updated theme clusters, ranking-based probabilistic generative models for each theme cluster is updated accordingly, which triggers the next round of clustering refinement. It is expected that the quality of clusters should be improved during this iterative update process since the similar sentences under new space will be grouped together and thus offers better attributes for further clustering.

Table 2 summarizes the whole process that generates sentence clusters in a given document set.

5. Theme-based summarization

5.1. Cluster number estimation

Our aim is to cluster sentences and select sentences in each cluster to form a summary. Note that the ranking-based sentence clustering framework requires a predefined cluster number K. To avoid exhaustive search for a proper cluster number for each document set, we employ the spectra approach introduced in [23] to predict the number of the expected clusters. Based on the sentence similarity matrix using the normalized 1-norm, for its eigenvalues δ_i ($i = 1, 2, \ldots, n$), the ratio $\varphi_i = \delta_{i+1}/\delta_2 (i \geqslant 1)$ is defined. If $\varphi_i - \varphi_{i+1} > 0.05$ and φ_i is still close to 1, then set $K = i + 1$.

5.2. Sentence extraction and redundancy removal

Once the sentence clusters are obtained, the summary sentences are then extracted from the original documents according to the ranks of the sentence clusters they belong to and their ranks within the assigned clusters.

The ranking score of each sentence cluster is formulated as

$$\gamma(C_k) = \frac{Score(C_k)}{\sum_{i=1}^{K} Score(C_k)} \tag{17}$$

Table 2
Generation process of sentence clusters.

Input: Document tri-type star graph G, Cluster number K and smoothing parameter λ
Output: Membership of each sentence
1: **Begin**:
2: Randomly partition sentences and generate initial theme-clusters from the original network according to these partitions, i.e. $\{C_k^0\}_{k=1}^K$
3: **repeat**
4: **foreach** theme-cluster $C_k \subseteq C$
5: Build ranking-based probabilistic generative model: $\{P(v|C_k^t)\}_{k=1}^K$
6: Calculate the posterior probabilities for sentence: $P(C_k^t|s)$
7: Adjust sentence cluster assignment according to the new measure defined by the posterior probabilities to each cluster.
8: **end**
9: **until** the cluster does not change significantly: $\{C_k^*\}_{k=1}^K = \{C_k^t\}_{k=1}^K = \{C_k^{t-1}\}_{k=1}^K$
10: **End**

where $Score(C_k)$ is formulated as the normalized cosine similarity between a theme cluster and the whole document set for generic summarization, or between a theme cluster and a given query for query-based summarization. $Score(C_k) \in [0, 1]$ and $\sum_{k=1}^{K} Score(C_k) = 1$.

The ranking score of each sentence within its assigned theme cluster is formulated as

$$\eta(s_i) = \frac{sim(s_i, C_k)}{\sum_{i=1}^{|C_k|} s_i} \tag{18}$$

where $|C_k|$ is sentence number of cluster C_k.

The summaries are then generated by choosing the highest ranked sentence from the highest ranked theme cluster to lowest ranked theme cluster, then the second highest sentences from theme clusters in descending order of rank, and so on.

Meanwhile, in multi-document summarization, the number of documents to be summarized can be very large. This makes information redundancy appears to be more serious in multi-document summarization than in single-document summarization. Redundancy control is necessary. Two popular techniques for avoiding redundancy in summarization are Maximal Marginal Relevance (MMR) [24] and clustering [25,26]. In MMR, the determination of redundancy is based mainly on the textual overlap between the sentence that is about to be added to the output and the sentences that are already in the generated summary text. MMR has been modified by many researchers [27,28]. On the other hand, clustering offers an alternative that the summarization system clusters the input textual units before starting the selection process. This step allows analyzing one or a few number of representative units from each cluster instead of all textual units.

We apply a simple yet effective way to choose summary sentences, which is a modified MMR-like approach. That is, at the beginning, we choose the first sentence from the ranking list into the summary. Then we examine the next one and compare it with the sentence(s) already included in the summary. Only the sentence that is not too similar to any sentence in the summary (i.e., the cosine similarity between them is lower than a threshold) is selected into the summary. This process is repeated until the length of sentences in the summary reaches the length limitation. In our experiment, the threshold is set to 0.9.

6. Experiment and evaluation

6.1. Experiment setup

We use surface term matching as the baseline of the experiments. We apply WordNet, LSA and context-based matching to test the influence of sentence representation for sentence clustering. For WordNet-based matching, and differently from that discussed earlier, we identify the WordNet synonym relations from the words in a given document set and then merge them into concepts to produce a sentence-by-concept matrix whose dimension is smaller than that of the original sentence-by-term matrix. LSA-based and context-based matching are the same as described earlier. For LSA-based matching, some studies have shown good results with a low dimensionality [29]. They declared that it is enough to use up to 10 dimensions (i.e., topics); therefore, we also set the number of topics to 10 in our experiments. Beside, we use three-level co-clustering frameworks [19], i.e. interactive co-clustering framework and integrated co-clustering framework, to test performance of sentence clusters. We further apply the proposed ranking-based sentence clustering framework to test whether sentence-clustering results can be improved when the ranking distribution of terms and documents are used as features of sentences.

We conduct a series of experiments on the DUC 2004 generic multi-document summarization dataset and the DUC2007 query-based multi-document summarization dataset. According to task definitions, systems are required to produce a concise summary for each document set (without or with a given query description) and the length of summaries is limited to 665 bytes in DUC 2004 and 250 words in DUC2007. In all the experiments, documents are pre-processed by segmenting sentences and splitting terms. Stop terms are removed and the remaining terms are stemmed using Porter Stemmer.[2]

6.2. Evaluation methods

We define cluster quality to evaluate the performance of the generated clusters. We also use ROUGE [30] to evaluate the performance of the generated summarization based on the generated clusters.

6.2.1. Intrinsic cluster quality evaluation

In order to evaluate the sentence cluster quality, we need to construct a sentence graph model $G_S = (S, E')$ at first, where $S = \{s_1, s_2, \ldots, s_n\}$ is the set of vertices representing sentences in document sets, E' is the set of edges connecting two sentences, every edge in E' is associated with a weight measuring the cosine similarity between the corresponding two sentences. Newman and Givan [31] define modularity measure Q as follow:

$$Q = \sum_i (e_{ii} - a_i^2) = Tr\ \mathbf{e} - ||\mathbf{e}||^2 \tag{19}$$

[2] http://www.tartarus.org/~martin/PorterStemme.

where the matrix **e** is a $K \times K$ symmetric matrix whose element e_{ij} is the fraction of all edges in the network that link vertices in community i to vertices in community j (K is the number of communities in the network). $a_i = \sum_j e_{ij}$ represents the fraction of edges that connect to vertices in community i. $Tre = \sum_i e_{ii}$ and $||\mathbf{x}||$ is the sum of the elements of the matrix X. The traditional modularity measure is applied in disconnected graph, while the constructed sentence graph is connected graph. Thus we modify the elements of the matrix **e**, i.e., e_{ij}, to be the fraction of all edges' weight in G_S that connect vertices in cluster C_i to vertices in cluster C_j. The generated sentence clusters are then evaluated by the modified modularity measure.

6.2.2. Extrinsic summarization evaluation

Our final aim is to generate more accurate summarization. We use ROUGE evaluation toolkit to evaluate the generated summarization, which has long been adopted by DUC for automatic summarization. It measures summary quality by counting overlapping units between system-generated summaries and human-written reference summaries. We report three common ROUGE scores in this report, namely ROUGE-1, ROUGE-2 and ROUGE-SU4, which base on Uni-gram match, Bi-gram match and Skip-Bi-gram match, respectively.

6.3. Experimental results

6.3.1. Parameter setting

The purpose to conduct this set of experiments is to examine and fix the settings of the parameter λ that is responsible for avoiding zero probabilities. We tune the values of λ from 0 to 1 with stepsize 0.1. The cluster quality values are presented in Table 3.

We can see from Table 3 that cluster quality of the proposed framework is very stable when λ ranges from 0.2 to 0.8, and the best result is obtained at $\lambda = 0.6$.

So we use $\lambda = 0.6$ in the following experiments.

6.3.2. Performance of sentence clustering based on different similarity measures

We set $\alpha = 0.75$, $\delta = 0.7$, $\mu = 0.1$, $\eta = 0.2$ for the two co-clustering frameworks which have been used in [19]. The evaluation of cluster quality based on the different similarity measures on the DUC2004 and the DUC2007 datasets are shown in Tables 4 and 5.

From the Tables 4 and 5, sentence clustering based on the cosine similarity shows the poorest cluster quality. This corresponds to the sense that the vector representation of the sentence is to be a very sparse representation. Thus it does not provide enough contexts for computing cosine similarity of the two sentences.

Clustering using enriched sentence representation can overcome the above weakness. However, the synonym of terms extracted from WordNet in the given document set is limited, meanwhile some terms do not exist in WordNet and these terms are usually named entities and can carry important information for summarization. In our experiments, about 13% of the terms in DUC 2004 dataset and 15% of the terms in DUC 2007 dataset do not exist in WordNet. Besides, WordNet provides very general domain knowledge about terms. These factors influence the accuracy of sentence clustering to some extent.

LSA can be used to construct specific, corpus-driven knowledge about terms, so the performance of LSA is better than that of WordNet. Although Islam and Inkpen [14] claimed that each dimension of the singular vector space captures a base latent semantics of the given document set and that each sentence in the document is jointly indexed by the base latent semantics in this space, negative values in some of the dimensions generated by the SVD make the above explanation less meaningful. Thus LSA cannot capture the exact semantic meaning of each sentence, which may reduce the accuracy of sentence clustering result either.

Sentence clustering based on context enrichment achieves better performance than that based on concept enrichment, which corresponds to the fact that more related original information can assist expressing sentence's meaning.

Table 3
Performance of the proposed framework with different λ values on the DUC2004 and DUC2007 datasets.

λ	Q	
	DUC2004	DUC2007
0	0.538	0.619
0.1	0.552	0.637
0.2	0.577	0.660
0.3	0.578	0.659
0.4	0.577	0.659
0.5	0.578	0.660
0.6	**0.579**	**0.661**
0.7	0.578	0.659
0.8	0.578	0.660
0.9	0.549	0.638
1	0.533	0.621

Table 4
Cluster quality evaluation on the DUC2004 dataset.

	Q
Ranking-based	0.579
Interactive	0.573
Integrated	0.551
Context-based	0.430
LSA-based	0.417
WordNet-based	0.358
Word-based	0.241

Table 5
Cluster quality evaluation on the DUC2007 dataset.

	Q
Ranking-based	0.661
Interactive	0.658
Integrated	0.633
Context-based	0.487
LSA-based	0.452
WordNet-based	0.373
Word-based	0.315

Two co-clustering frameworks, i.e., integrated co-clustering framework and interactive co-clustering framework, show better performance than sentence representation reformation, which confirms that the terms and documents are used as independent text objects is better than terms are used as features of sentences or sentences are used as components of documents.

Ranking-based clustering framework shows the best performance, it further presents ranking distribution of documents and terms can help generate more accurate sentence clusters.

We have implemented the traditional MMR approach proposed by Carbonell and Goldestein [32] and our modified MMR-like approach for redundancy control. The ROUGE values based on the different similarity measures on the DUC2004 and DUC2007 datasets are shown in Tables 6 and 7

From Tables 6 and 7, it can be observed that the two approaches have very similar ROUGE results. Considering the advantage of its easy implementation and high computational efficiency, we decided to apply our proposed MMR-like redundancy control approach in all remaining experiments. The two tables also show that the performance of each approach is in accordance with that of cluster quality. It further indicates the fact that good clusters bring about good summarization.

6.3.3. T-test evaluation

The significance of the improvement is always of concern. Based on it, we further conduct the paired *t*-test evaluation using ROUGE-2 scores, the primary DUC evaluation criterion, on all 50 DUC2004 document set and 45 DUC 2007 document sets. The hypothesis here is that "the first approach is equal to or inferior to the second one in ROUGE-2" and the significance level is 5%.

Table 6
Summarization ROUGE evaluation on the DUC2004 dataset.

	ROUGE-1	ROUGE-2	ROUGE-SU4
Ranking-based MMR-like	0.37878	0.09357	0.13253
Ranking-based traditional MMR	0.34875	0.09356	0.13251
Interactive MMR-like	0.37872	0.09203	0.13257
Interactive traditional MMR	0.37870	0.09202	0.13256
Integrated MMR-like	0.37579	0.08697	0.12875
Integrated traditional MMR	0.37576	0.08694	0.12873
Context-based MMR-like	0.35346	0.07042	0.11553
Context-based traditional MMR	0.35344	0.07041	0.11551
LSA-based MMR-like	0.34583	0.06679	0.11096
LSA-based traditional MMR	0.34581	0.06677	0.11095
WordNet-based MMR-like	0.34187	0.06345	0.10843
WordNet-based traditional MMR	0.34185	0.06344	0.10840
Word-based MMR-like	0.33732	0.06038	0.10081
Word-based traditional MMR	0.33731	0.06035	0.10799

Table 7
Summarization ROUGE evaluation on the DUC2007 dataset.

	ROUGE-1	ROUGE-2	ROUGE-SU4
Ranking-based MMR-like	0.44221	0.12618	0.17802
Ranking-based traditional MMR	0.44220	0.12616	0.17800
Interactive MMR-like	0.44219	0.12615	0.17799
Interactive traditional MMR	0.44217	0.12614	0.17796
Integrated MMR-like	0.44121	0.12485	0.17492
Integrated traditional MMR	0.44119	0.12483	0.17491
Context-based MMR-like	0.39875	0.09973	0.15102
Context-based traditional MMR	0.39873	0.09970	0.15100
LSA-based MMR-like	0.39521	0.09879	0.14915
LSA-based traditional MMR	0.39519	0.09877	0.14913
WordNet-based MMR-like	0.39015	0.09739	0.14503
WordNet-based traditional MMR	0.39014	0.09737	0.14500
Word-based MMR-like	0.38531	0.08643	0.13311
Word-based traditional MMR	0.38529	0.08641	0.13310

Table 8
T-test evaluation on DUC2004 and DUC2007.

	p	
	DUC2004	DUC2007
WordNet-based vs. Word-based	0.02075	0.02793
LSA-based vs. WordNet-based	0.02197	0.02651
Context-based vs. LSA-based	0.02178	0.02975
Integrated-based vs. Context-based	0.03677	0.03416
Interactive-based vs. Integrated-based	0.03889	0.03675
Ranking-based vs. Interactive-based	0.03904	0.03691

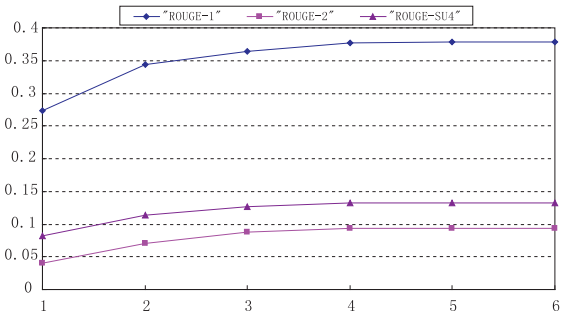

Fig. 4. Trends of ROUGEs with increased iteration numbers on DUC2004 dataset.

Table 9
Comparison with DUC participating systems on the DUC2004 dataset.

	ROUGE-1	ROUGE-2	ROUGE-SU4
Ranking-based clustering	0.37878	0.09357	0.13253
System 65	0.37816	0.09147	0.13178
System 35	0.37076	0.08335	0.12733
System 104	0.37045	0.08527	0.12763
Coverage	0.34729	0.06983	0.10498

The *P*-values presented in Table 8 suggests that all the hypotheses are rejected, which means the first approach is superior to the second one. The evaluation results further confirm that our analysis is correct.

6.3.4. Further analysis on cluster quality's improvement with ranking-based clustering

The aim of ranking-based clustering framework is to generate more accurate sentence clusters by iteratively refining new measure defined by the posterior probabilities to each theme cluster. Fig. 2 plots the values of Q (the sentence cluster

Table 10
Comparison with DUC participating systems on the DUC2007 dataset.

	ROUGE-1	ROUGE-2	ROUGE-SU4
Ranking-based clustering	0.44221	0.12618	0.17802
System 15	0.44100	0.12392	0.17501
System 4	0.43005	0.11809	0.16789
System 24	0.44922	0.11757	0.17427
Coverage	0.40378	0.10443	0.14963

quality) in each iteration of the ranking-based clustering on the DUC 2004 dataset. The increase of Q indicates the improvement of the cluster quality.

While Q directly evaluates the quality of the generated clusters, we are also interested to know whether the improved quality of clusters can further enhance the quality of sentence ranking and thus consequently raise the performance of summarization. Therefore, we evaluate the ROUGEs in each iteration of ranking-based clustering as well. Fig. 4 illustrates the increase of ROUGE-1, ROUGE-2 and ROUGE-SU4 results on the same dataset mentioned above. The figures demonstrate the significant role of the proposed ranking-based clustering framework in summarization.

6.3.5. Comparison with DUC summarization systems

Next, we compare the proposed ranking-based clustering framework with a coverage baseline, which takes the first sentence in the first document, the first sentence in the second document and so on until it had a summary of 665 bytes for DUC2004 dataset and 250 words for DUC2007 dataset. For comparison purpose, the ROUGEs results of top three DUC systems participating in DUC2004 and DUC2007 are also included. Tables 9 and 10 present the comparison results.

The advantages of the proposed ranking-based clustering framework are clearly demonstrated in the above tables. They produce very competitive results and significantly outperform the coverage baseline. More important, it is ahead of the best system in DUC2004 and DUC2007. We contribute to ranking distributions of terms and documents can help generate more accurate sentence clusters.

7. Conclusion

Sentence clustering relies heavily on the sentence similarity methods, but the sentences with similar meaning always share few common words, so traditional bag-of-words cosine similarity is no longer suitable for measuring sentence similarity. In this paper, we first define two different ranking functions in a tri-type document star graph constructed from the given document set. Then, we build ranking-based probabilistic generative model and calculate the posterior probabilities for each sentence. Sentences then are reassigned to the nearest cluster under the new measure space to improve clustering. As a result, quality of sentence clustering is enhanced. Experimental results show that the ranking-based clustering framework is able to generate more reasonable sentence clusters and in turn lead to better summarization performance. In the future, we will focus on the influence of other proper context definition besides words and documents on sentence clustering in order to further enhance performance of sentence clustering.

Acknowledgements

The work described in this paper was partially support by National Natural Science Foundation of China (Project Nos. 61202188, 61303125, 61303226), Central Universities under Grant (Project Nos. Z109021109, Z109021108) and scientific research fund for the junior teachers in Northwest Agriculture and Forestry University (Nos. Z109021104, Z109021105).

References

[1] C.D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, 1999.
[2] K.S. Jones, Automatic summarising: the state of the art, Inf. Process. Manage. 43 (2007) 1449–1481.
[3] D.R. Radev, E. Hovy, K. Mckeown, Introduction to the special issue on summarization, Comput. Linguist. 28 (2002) 399–408.
[4] R.M. Aliguliyev, A new sentence similarity measure and sentence based extractive technique for automatic text summarization, Expert Syst. Appl. 36 (2009) 7764–7772.
[5] Allan, J., Bolivar, A., Wade, C., 2003. Retrieval and novelty detection at the sentence level, in: Proceedings of the 26th Annual International Conference on Research and Development in, Information Retrieval (SIGIR'03), pp. 314–321.
[6] C.D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, 1999.
[7] D. Metzler, Y. Bernstein, W. Croft, A. Moffat, J. Zobel, Similarity measures for tracking information flow, in: Proceedings of the 14th International Conference on Information and Knowledge Management (CIKM'05), 2005, pp. 517–524.
[8] F. Wei, W. Li, Q. Lu, Y. He, Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization, in: Proceedings of the 31th Annual international conference on Research and Development in Information Retrieval (ACM SIGIR'08), 2008, pp. 283–290.
[9] G. Kumaran, J. Allan, Text classification and named entities for new event detection, in: Proceedings of the 27th Annual International Conference on Research and Development in, information Retrieval (SIGIR'03), 2004, pp. 297–304.
[10] K.M. Hammouda, M.S. Kamel, Efficient phrase-based document indexing for web document clustering, IEEE Trans. Knowl. Data Eng. 16 (10) (2004) 1279–1296.
[11] L. Zhao, L. Wu, X.J. Huang, Fudan University at DUC 2006. Ducument Understanding Conferences 2006 (DUC'06), 2006.

[12] S. Banergee, K. Ramanathan, A. Gupta, Clustering short texts using Wikipedia, in: Proceedings of the 30th Annual International Conference on Research and Development in Information Retrieval (SIGIR'07), 2007, pp. 787–788.
[13] T.K. Landauer, P.W. Foltz, D. Laham, An introduction to latent semantic analysis, Discour. Process. 25 (2&3) (1998) 259–284.
[14] A. Islam, D. Inkpen, Semantic text similarity using corpus-based word similarity and string similarity, ACM Trans. Knowl. Discov. Data 2 (2) (2008) 1–25.
[15] P. Foltz, W. Kintsch, T. Landauer, The measurement of textual coherence with latent semantic analysis, Discour. Process. 25 (2&3) (1998) 285–307.
[16] X.Y. Cai, W.J. Li, Context-sensitive manifold ranking approach for query-focused multi-document summarization, Lect. Notes Comput. Sci. (2010) 27–38.
[17] M. Mitra, A. Singhal, C. Buckley, Automatic text summarization by paragraph extraction, in: Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization, 1997.
[18] T. Strzalkowski, J. Wang, B. Wise, A robust practical text summarization system, in: Proceedings of 12th AAAI Conference on Artificial Intelligence (AAAI'98), 1998, pp. 2630–2639.
[19] X.Y. Cai, W.J. Li, Enhancing sentence-level clustering with integrated and interactive frameworks for theme-based summarization, J. Am. Soc. Inform. Sci. Technol. 62 (10) (2011) 2067–2082.
[20] Z. Nie, Y. Zhang, J.R. Wen, W.-Y. Ma, Object-level ranking: bringing order to web objects, in: WWW'05, 2005, pp. 567–574.
[21] Y.Z. Sun, Y.T. Yu, J.W. Han, Ranking-based clustering of heterogeneous information networks with star network schema, in: KDD'09, 2009, pp. 797–805.
[22] C. Zhai, J.D. Lafferty, A study of smoothing methods for language models applied to information retrieval, ACM Trans. Inf. Syst. 22 (2) (2004) 179–214.
[23] J. LiW, K. NgW, Y. Liu, K.L. Ong, Enhancing the effectiveness of clustering with spectra analysis, IEEE Trans. Knowl Data Eng. 19 (7) (2007) 887–902.
[24] J.G. Corbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: Proc. 21st SIGIR Conf., 1998, pp. 335–336.
[25] K.R. Mckeown, J.L. Kalvans, V. Hatzivassiloglou, R. Barzilay, E. Eskin, Towards multi-document summarization by reformulation: progress and prospects, in: Proc. 13th AAAI Conf., 1999, pp. 121–128.
[26] D.R. Radev, H.Y. Jing, M. Stys, D. Tam, Centroid-based summarization of multiple documents, Inform. Process. Manage. 40 (6) (2004) 919–938.
[27] L. Antiqueris, O.N. Oliveira, L.F. Costa, M.G. Nunes, A complex network approach to text summarization, Inform. Sci. 175 (5) (2009) 297–327.
[28] E. Filatova, V. Hatzivassiloglou, Event-based extractive summarization, in: Proc. 42nd ACL Conf. 2004, pp. 104–111.
[29] J. Steinberger, M. Křišťan, LSA-based multi-document summarization, in: Proceedings of 8th International Workshop on Systems and Control'07, 2007.
[30] C.Y. Lin, E. Hovy, The automated acquisition of topic signature for text summarization, in: Proceedings of 23rd International Conference on Computational Linguistics (COLING'2000), 2000, pp. 495–501.
[31] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 8577–8582.
[32] J.G. Corbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21th Annual International Conference on Research and Development in Information Retrieval (SIGIR'98), 1998, pp. 335–336.
[33] C. Burgess, K. Livesay, K. Lund, Explorations in context space: words, sentences, discourse, Discour. Process. 25 (2–3) (1998) 211–257.
[34] J.L. McClelland, A.H. Kawamoto, Mechanisms of sentence processing: assigning roles to constituents of sentences, in: D.E. Rumelhart, J.L. McClelland, the PDP Research (Eds.), Parallel Distributed Process, vol. 2, MIT Press, 1986, pp. 272–325.
[35] V. Hatzivassiloglou, J. Klavans, E. Eskin, Detecting similarity by applying leaning over indicators, in: Proc. 37th Ann. Meeting of the Assoc. for Computational Linguistics, 1999.
[36] D. Jurafsky, J.H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall, 2000.