Knowl Inf Syst (2012) 31:527–545
DOI 10.1007/s10115-011-0437-x

REGULAR PAPER

SumCR: A new subtopic-based extractive approach for text summarization

Jian-Ping Mei · Lihui Chen

Received: 1 September 2010 / Revised: 1 June 2011 / Accepted: 18 July 2011 /
Published online: 6 August 2011

Abstract In text summarization, relevance and coverage are two main criteria that decide the quality of a summary. In this paper, we propose a new multi-document summarization approach SumCR via sentence extraction. A novel feature called *Exemplar* is introduced to help to simultaneously deal with these two concerns during sentence ranking. Unlike conventional ways where the relevance value of each sentence is calculated based on the whole collection of sentences, the *Exemplar* value of each sentence in SumCR is obtained within a subset of similar sentences. A fuzzy medoid-based clustering approach is used to produce sentence clusters or subsets where each of them corresponds to a subtopic of the related topic. Such kind of subtopic-based feature captures the relevance of each sentence within different subtopics and thus enhances the chance of SumCR to produce a summary with a wider coverage and less redundancy. Another feature we incorporate in SumCR is *Position*, i.e., the position of each sentence appeared in the corresponding document. The final score of each sentence is a combination of the subtopic-level feature *Exemplar* and the document-level feature *Position*. Experimental studies on DUC benchmark data show the good performance of SumCR and its potential in summarization tasks.

Keywords Text summarization · Clustering · Subtopic · Sentence extractive · Sentence position

1 Introduction

The explosive growing of on-line documents in recent years attracts a lot of interests in multi-document summarization (MDS), which automatically generates a single short summary for a set of documents related to the same topic. According to the content, summaries can be either task-focused or generic. A task-focused, or topic-driven, or query-oriented summary is tailored for the requirement of a particular group of users by extracting information from

J.-P. Mei · L. Chen (✉)
Division of Information Engineering, School of Electrical and Electronic Engineering,
Nanyang Technological University, 50 Nanyang Avenue, Singapore, Republic of Singapore
e-mail: elhchen@ntu.edu.sg

the documents that is most related to a given topic. Differently, a generic summary tries to present all the main points conveyed by the input documents without any specific external instruction or requirement.

A summary is typically generated with two main categories of techniques, called extraction and abstraction [16,17,23]. Extractive summarization simply extracts salient information, such as sentences, from the input documents and "put them together" to form summaries. Although summaries generated in this way may lack of coherence, extractive approaches are still popular in nowadays as they are low cost and easy to be applied to general domains. Aiming to produce grammatical coherent summaries, abstractive summarization creates summaries by synthesizing and rewriting sentences based on contextual and linguistic understanding, which is heavily dependent on deep analysis and language generation techniques. Rather than using a single technique alone, some researchers are interested in regeneration as a post-process for extractive summaries, i.e., make pruning or revision based on extractive summaries [23,31]. In this paper, we focus our efforts on multi-document summarization through sentence extraction. A large majority of sentence extractive-based summarization approaches follow the classic framework proposed by [10]. In this framework, a summary is formed through sentence scoring, where a score is assigned to each sentence as a combination value of weights of several predefined features. The critical part in this process is to define and weight each of the features.

Due to the existence of information overlapping and repeating in the input documents, simply selecting sentences with high global relevance scores may cause redundancy, i.e., some of those selected sentences may provide the same or similar information. This problem becomes more critical in multi-document summarization. Since as a condensed version of the original input documents, the overall coverage of a summary is as important as the relevance or representativeness of each individual sentence included in the summary. Some researchers began to look for explicit ways to reduce redundancy and thus enhance the diversity of the content of the summary produced. [7] proposed the Maximal Marginal Redundancy (MMR) principle to reduce repeating information by selecting sentences to be not too similar to each other. The reranker applied in MEAD [33] also follows the MMR principle.

Other than the redundancy problem, different subtopics or concepts always exist. In other words, multiple concepts or several aspects of a topic is probably involved in description of a topic. Traditional statistical features obtained based on the whole input corpus may not be able to capture some of the underlying subtopics, and thus lose some degree of completeness of the summary content. Machine learning approaches are used in many recent studies to capture subtopics [2,8,14,30,35,37]. In approaches such as [2,30,37], sentences segmented from the document set are first grouped into clusters so that sentences in the same cluster are more similar than those in different clusters. Based on the sentence clusters, the most representative sentences within each of the clusters are picked out to form the summary. Selecting sentence from different clusters may help to increase the diversity and the coverage, and at the same time alleviate the redundancy problem. Some other approaches by [8,14,35] using sentence-level probabilistic topic modeling where each subtopic is a model with predefined distribution and each sentence is assumed to be generated with different probabilities from those models. The effectiveness of subtopic-based approaches depends on the existence of the subtopic structure. When there is no obvious subtopics in the dataset, clustering-based approaches may become less effective, and hence causes the degradation of the quality of summaries produced.

In this paper, we propose a novel sentence extractive summarization SumCR using a new subtopic-level feature *Exemplar* and a document-level feature *Position*. The *Exemplar* feature is defined based on a recently developed clustering method called PFC [25]. Unlike model-based approaches, which require each document to be treated as “a bag of words”, PFC clusters sentences into groups by making use of similarities of pairs of sentences. Taking this general representation form of the input data, PFC provides a convenient interface for advanced sentence similarity measure technique to be further integrated. After clustering, sentences that are similar to each other in content are grouped into the same cluster where each cluster corresponds to a subtopic or a topic area. An important reason for using the PFC clustering approach for sentence grouping is that it simultaneously generates the groups of sentences as well as the degree of representativeness of each sentence in the associated group, called prototype weight. Based on the output of PFC, i.e., sentence clusters and prototype weights, the *Exemplar* weights are computed for each of the sentences. This *Exemplar* is a novel feature that plays a critical role in sentence ranking in our approach. To maintain a good performance on various datasets, we score sentence by combining the weights of *Exemplar* and another feature *Position*, which reflects the location of the sentence in the corresponding document. With the combination of subtopic-level feature and document-level feature, SumCR aims to produce a summary with a high relevance as well as a wide coverage and little redundancy. By further considering the closeness to a user-given query, SumCR is extended to task-focused summarization.

We conduct experiments for both generic and task-focused multi-document summarization on benchmark data DUC2004, DUC2005, and DUC2006, respectively. The summarization results are evaluated by ROUGE-2 and ROUGE-SU4 scores. The experimental results show that with a simple similarity measure and without using any sophisticated query processing procedures, the proposed approach outperforms all the state-of-the-art approaches in the generic summarization task on DUC2004, and the results of SumCR for task-focused summarization on DUC2005 and DUC2006 rank at 4th or 5th among over 30 systems participated in DUC and several other recently developed approaches.

The rest of the paper is arranged as follows: in Sect. 2, we review some existing work on summarization which are most related to the proposed approach. The details of the proposed summarization approach SumCR are presented in Sect. 3, where we first give an over view of the new approach and then discuss the details of each component. In Sect. 4, experiments on benchmark data are conducted and the results are discussed. Finally, conclusions of this study are given in Sect. 5.

2 Related work

The framework established by [10] is a classic paradigm in extractive summarization that continues to influence the research work on this topic today. Four features are used in it, namely *cue words*, *title words*, *key words*, and *sentence location*. The overall score of each sentence is obtained through a linear combination of the weights of the four features. A summary is then formed by choosing a needed number of sentences with the largest overall scores. Quite a lot of subsequent researches on extractive summarization, including some notable summarization systems, such as SUMMARIST [15], MEAD [33], and SumBasic [28], follow this framework with improvements made by expanding the feature set and using more sophisticated methods for weighting different features. Other than introducing a number of new features, [15] studied the best positional feature for sentence extraction. [28] studied the usefulness of a particular feature—frequency in their summarization system SumBasic, and they show that good quality summaries can be produced based on frequency alone. By further incorporating the relevance of each sentence to the query, SumBasic has been extended to a task-focused summarization in [36]. The MEAD proposed by [33] is one successful system for large-scale multi-document summarization where a core feature *Centroid* is derived to measure the centrality of a sentence. The *Centroid* score of each sentence is computed based on its similarity to the centroid, a virtual sentence consisting of important words.

Other than following the classic framework, other techniques from related fields have also been employed to document summarization including graph-based approaches and machine learning approaches. Graph-based methods such as LexPageRank [11], TextRank [26] are proposed to rank sentences using graph ranking algorithms that are previously used for social network analysis and Web structure analysis. In these approaches, each text unit, e.g., sentence, is a node or vertex on a graph and the weight of the edge between two nodes is the similarity between two sentences. Sentences are then weighted recursively by taking account of global information from the graph. One important issue in graph-based approach is how to construct the graph. The recently proposed document-sensitive graph model [38] is shown to be better than other graph models where the document boundary information is not considered. Different machine learning approaches have been applied for summarization, including supervised approaches, such as [3,18], and unsupervised approaches, such as Latent semantic analysis [13], probabilistic generative model [14,35], Hidden Markov Model [9], Conditional Random Field [34], Non-negative Matrix Factorization [19], and other clustering-based approaches [2,30,37]. A recent work by [8] uses both unsupervised and supervised learning to build a generative model for cluster discovery and a regression model for inference. In [22], the best summary is defined to be the one which has the minimum information distance to the entire document set.

As observed in [5], the content of documents to be summarized always contains multiple subtopics describing the same issue from different aspects. Traditional approaches that score sentence with global features extracted based on the whole document set is unable to discover those subtopics. Recent studies, such as [8,14,35], build probabilistic models for each of the subtopics on sentence level. The model distribution for each sentence shows how possible this sentence is generated from each of the models. In [4], other than using Latent Dirichlet Allocation (LDA) to discover subtopics in the document set, Singular Value Decomposition (SVD) is further used to find the sentences that best represent these subtopics. Although model-based approaches have a solid mathematical foundation, these approaches are based on the “a bag of words” representation of each sentence, which only preserves the basic statistical information of the original sentence. Other than model-based approaches, clustering-based approaches also aim to produce subtopic-level summaries. Clustering is a useful unsupervised machine learning technique that has been successfully applied in information retrieval for various purposes, such as extracting common emotions from blogs [12] and summarizing data streams [1]. In clustering-based summarization, sentences are first grouped into clusters and then representative sentences of each of the clusters are selected to form the summary. Each sentence cluster corresponds to a subtopic, and such a cluster-based selection aims to help to increase the diversity and the coverage of the summary, which also alleviates the redundancy problem.

In clustering-based summarization systems, the selection of a particular clustering approach to generate reasonable sentence clusters and the proper definition of a feature by making use of the clustering results are two related issues that contribute to the effectiveness of the summarizer. To group sentences into subgroups, [30] use a modified k-mean clustering with sentences being represented as word vectors, while [37] uses symmetric non-negative factorization of the sentence similarity matrix (SNMF). A good reason to use clustering approach with pairwise similarity matrix as input is that any sentence similarity measure technique can be easily adopted and integrated into the system, and the final summarization result may be improved with a more accurate sentence similarity measure, e.g., in SNMF [37], the quality of summaries are shown to be improved when semantic sentence

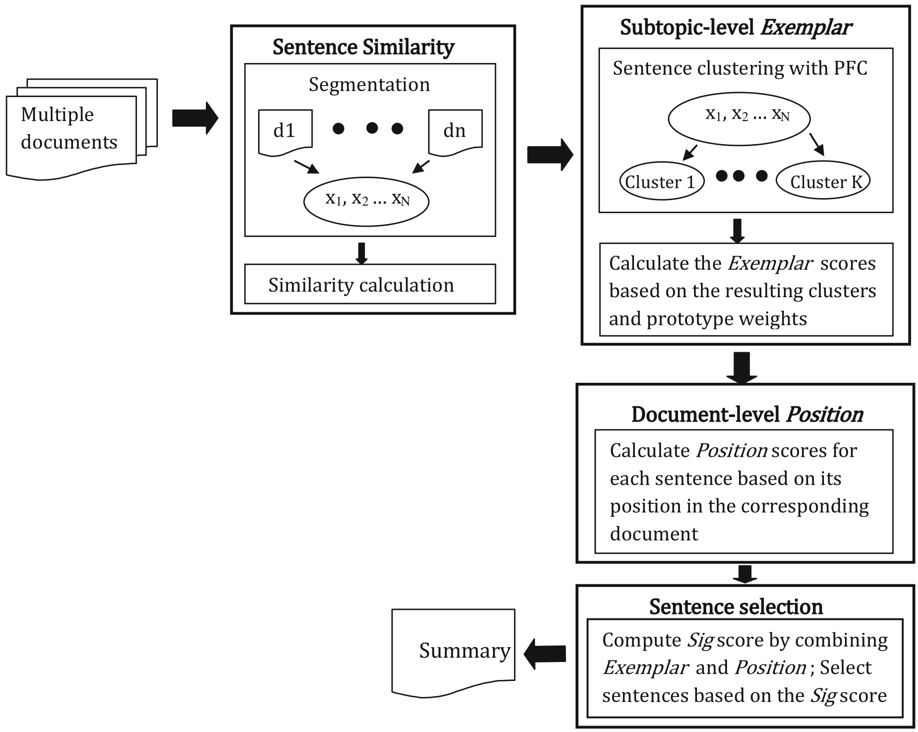

Fig. 1 Overview of the proposed SumCR summarizer

similarity is used to replace the simple keyword-based one. However, in SNMF, it is unable to define the cluster center explicitly given the similarity matrix. Therefore, unlike in [30], where both closeness to the cluster center and closeness to the sentences in the same cluster are considered in defining the subtopic-based feature, only the later is used in SNMF [37]. To obtain an effective subtopic-level feature for extractive summarization, it is important to have a clustering approach that is able to produce sentence clusters with a high quality, provide useful measurement on the within-cluster importance of each sentence, and is not limited to a specific representation model of the text unit. In the next section, we will show that the clustering approach PFC is a good candidate with these favored properties for summarization, based on which we can introduce a novel subtopic-level feature called *Exemplar* and hence develop a new summarizer SumCR.

3 The proposed method: sumCR

We first present an overview of the core SumCR system for generic multi-document summarization. It follows by the detailed introduction to each of the components. An extended version of SumCR will also be given for task-focused summarization.

3.1 Overview

The overall system structure of the proposed SumCR is shown in Fig. 1. Given a set of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ related to the same topic, SumCR produces a short summary for the document set with several procedures given as follows:

Compute the sentence similarity matrix: First, each of the documents is segmented into sentences to get a pool of sentences $\mathcal{X} = \{x_i, x_2, \ldots, x_N\}$, where N is the total number of sentences from n documents. After that, the similarity matrix $S_{N\times N}$ recording similarities between every pair of sentences is calculated or estimated with any proper similarity measure techniques.

Calculate the *Exemplar* score: Once the similarity matrix is obtained, we perform the fuzzy clustering PFC [25] to group sentences into subsets or clusters. At the same time, prototype weight of each sentence with respect to a cluster is also produced at the end of the clustering process. The *Exemplar* weight of a sentence considers both the closeness to sentences in the same cluster and the prototype weight of this sentence in the cluster it belongs to.

Calculate the *Position* score: The *Position* score decreases with the sequence of the sentence appears in the document, i.e., the *Position* value of the first sentence is the largest, while the last sentence is the smallest.

Form the summary: After the weights of *Exemplar* and *Position* are calculated, sentences are finally scored as the combination of the two. Sentences are ranked with their scores in descendent order and those top ones are selected to form the summary. The pairwise similarities of sentences being selected need to be smaller than a given threshold.

The core part of SumCR is the weighting of the *Exemplar* feature based on the clustering result of PFC, which is a similarity-based soft clustering approach. A unique property of PFC that differentiates it from others is that it produces prototype weights of each sentence together with the sentence clusters. The prototype weight reflects within-cluster representativeness. Therefore, it provides an accurate measure of the relevance of each sentence in a subtopic it is most related to. With PFC, our proposed summarizer SumCR maintains the advantage of conveniently integrating any techniques for measuring sentence similarity, but also makes use of the prototype weight for producing a more effective subtopic-level feature *Exemplar*. Next, we present the details of the PFC approach and more discussions on its important properties, which make PFC a favorable choice in SumCR.

3.2 Fuzzy clustering with prototype weights: PFC

The PFC clustering approach is a new development of our research work in data clustering [25]. We first give the formulation and algorithm of PFC, and then explain why it is a better choice than other clustering approaches for text summarization. To be directly applied in SumCR, the PFC presented here is a similarity-based version, which is equivalent to the dissimilarity-based one reported in [25].

3.2.1 Problem formulation

Given the similarity matrix $S_{N\times N}$ with each element $s_{ij} \in S$ denoting the similarity between sentence x_i and x_j, the objective of PFC is to maximize the following criterion:

$$J_{\text{PFC}} = \sum_{c=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N} u_{ci} v_{cj} s_{ij} - \frac{T_u}{2}\sum_{c=1}^{K}\sum_{i=1}^{N} u_{ci}^2 - \frac{T_v}{2}\sum_{c=1}^{K}\sum_{j=1}^{N} v_{cj}^2 \tag{1}$$

subject to constraints

$$\sum_{c=1}^{K} u_{ci} = 1 \quad \text{for } 1 \le i \le N; \tag{2}$$

$$u_{ci} \geq 0 \quad \text{for} \;\; 1 \leq c \leq K, \quad 1 \leq i \leq N; \tag{3}$$

and

$$\sum_{j=1}^{N} v_{cj} = 1 \quad \text{for } 1 \leq c \leq K; \tag{4}$$

$$v_{cj} \geq 0 \quad \text{for} \;\; 1 \leq c \leq K, \quad 1 \leq j \leq N. \tag{5}$$

where K is the number of clusters, u_{ci} is the fuzzy membership representing how well x_i belongs to cluster c, and v_{cj} is the prototype weight which reflects how much x_j is weighed as a representative in cluster c. It can be seen that the objective function of PFC in Eq. (1) consists of three terms. The first term measures the total compactness of all clusters and controls the main direction of the clustering process; while the last two terms are the regularization of u and v used to prevent u and v from singular values. Parameters T_u and T_v control the tradeoff between the main term and the regularization terms. Our goal is to find all $u_{ci} \in U$ and $v_{cj} \in V$ to maximize J_{PFC} in Eq. (1) under the constraints in Eqs. (2–5).

Updating rules of u_{ci} and v_{cj} are derived as below with the Lagrange multiplier method

$$u_{ci} = \frac{1}{K} + \frac{1}{T_u}\left[\sum_{j=1}^{N} v_{cj} s_{ij} - \frac{1}{K}\sum_{f=1}^{K}\sum_{j=1}^{N} v_{fj} s_{ij}\right], \tag{6}$$

$$v_{cj} = \frac{1}{N} + \frac{1}{T_v}\left[\sum_{i=1}^{N} u_{ci} s_{ij} - \frac{1}{N}\sum_{q=1}^{N}\sum_{i=1}^{N} u_{ci} s_{iq}\right]. \tag{7}$$

Given the similarity matrix S, the number of clusters K, parameters T_u and T_v, the procedure to find an optimal fuzzy partition U, and optimal prototype weights V that maximizes J_{PFC} can be described as: start with an initial U, iteratively update V and U with Eqs. (7) and (6), respectively, in an alternating manner until convergence. It can be observed that negative values of U and V might occur during the updating process. Therefore, after updating at each iteration, negative entries of U (or V) are set to 0, and then re-normalization is performed accordingly to make U (or V) satisfy the summation constraints given in Eq. (2) [or Eq. (4)].

The alternately updating of U and V allows successively improvement in the quality of clusters by re-assigning clusters to sentences and re-estimating the representativeness of sentences in each cluster. The iteration terminates when the successive estimates of U are close enough. The same as other approaches using alternating optimization scheme, the PFC algorithm described above only converges to local solutions. A simple way to reduce the impact of local maximum is to run the algorithm for multiple times and pick the best one. More detailed discussions and analysis on PFC, including its superiorness in formulation, a complete version of the updating formulas, the convergence property, and time complexity could be found in [25].

3.2.2 Useful properties

Now, we highlight several important properties of PFC which make it a better choice than other clustering approaches with the application of summarization.

1. *Fuzzy clustering*: Compared with hard or crisp clustering like k-means, fuzzy clustering is more suitable to be used to generate sentence clusters. In hard clustering, overlap among clusters is not allowed, while in fuzzy clustering, each sentence can belong to

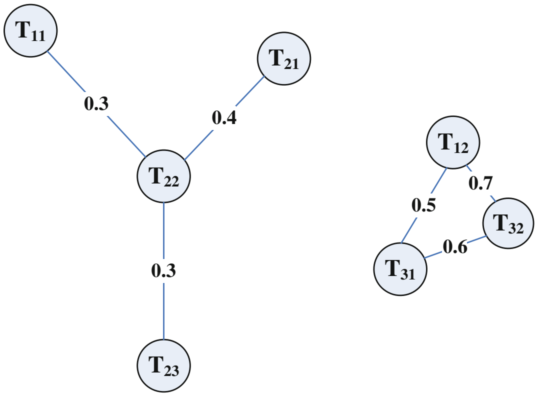

Fig. 2 An example data

more than one clusters with difference degrees of belonging. The use of fuzzy membership in clustering is able to capture the structure of the data more naturally because it is very likely that a single sentence is related to multiple subtopics to different degrees.

2. *Similarity based*: Unlike some other fuzzy clustering approaches which require the input sentences to be represented as vectors of a set of terms, PFC is designed for relational data, i.e., similarity or dissimilarity. This format of input enables the overall performance of SumCR to be benefited from the development of a more accurate similarity measure. In other words, any advanced technique, which gives a good similarity measure of sentences including statistical models and other models incorporating high-level NLP techniques, can be easily adopted and integrated into SumCR. There are already many studies, such as [27,32,37], that reported the success of using a good similarity measure to achieve final improvement in clustering and classification.
3. *Prototype weight*: The most distinctive characteristic of PFC compared with other existing (dis)similarity-based fuzzy clustering is that cluster-based representativeness called prototype weight is generated along with the clusters. When the input data are pairwise similarities, it is unable to define a cluster center as it did in a vector space; therefore, most existing (dis)similarity-based clustering approaches concentrate on obtaining the partitioning of the data without defining any variable to reflect the representativeness of objects in each cluster, such as SNMF in [37]. In PFC, we are interested in both partitions and cluster-based typicality. To achieve this, fuzzy membership together with the prototype weight are defined. With prototype weights, all sentences are able to represent each cluster to a certain extent. This enables PFC to capture the cluster structure more accurately and hence creates clusters with good qualities. Moreover, according to the definition, prototype weight provides a good measure of within-cluster representativeness of each sentence.

3.2.3 A simple example

Now, we give a simple example to illustrate the performance of PFC for data clustering. As shown in Fig. 2, we are now given a set of seven sentences $\mathcal{X} = \{x_1, x_2, \ldots, x_7\}$ segmented from three documents doc_1, doc_2, doc_3, and each $x_i = T_{pq}$ denotes the qth sentence of the pth documents. We use an undirected graph to represent the closeness among sentences. Each node represents a sentence and there is an edge between two nodes if their similarity as

Table 1 Clustering result of PFC on the example data

	T_{11}	T_{12}	T_{21}	T_{22}	T_{23}	T_{31}	T_{32}
u_1	0.6466	0.1298	0.7069	0.8243	0.6466	0.1567	0.1103
u_2	0.3534	0.8702	0.2931	0.1757	0.3534	0.8433	0.8897
v_1	0.1535	0	0.2276	0.4655	0.1535	0	0
v_2	0	0.3347	0	0	0	0.3004	0.3649

labeled on the edge is larger than a threshold. By setting $K = 2$, parameters $T_u = T_v = 1$, we get U and V produced by PFC as in Table 1.

From u_1 and u_2 in Table 1, it can be seen that $\{T_{11}, T_{21}, T_{22}, T_{23}\}$ have a larger membership in cluster C_1, while $\{T_{12}, T_{31}, T_{32}\}$ have a larger membership in cluster C_2. From v_1 and v_2, it shows that T_{22} in C_1 has a prototype weight which is significant larger than that of $\{T_{11}, T_{21}, T_{23}\}$; while the prototype weights of three sentences $\{T_{12}, T_{31}, T_{32}\}$ in C_2 are close to each other. From Fig. 2, it can be seen that T_{11}, T_{21}, and T_{23} are all similar to T_{22}, therefore, these four sentences may form a subtopic where T_{22} is the most representative one. The other three sentences form the second subtopic and have similar representativeness since any one of the three is similar to the other two. This example shows that the U and V produced by PFC well describe the data structure in different ways, i.e., U reflects the partitioning and V the representativeness within each cluster. Other similarity-based approaches, such as SNMF, only produce a partitioning of the data. In real applications, it is non-trivial to estimate the number of subtopics contained in a given document set even for human experts. Since for each cluster of sentences, we only select one representative sentence, the number of clusters K may be set to be close to the number of sentences to be chosen to form a summary with a required length. For example, we may first calculate the average length of sentences contained in the document set and then set K as the number of sentences with this average length in order to form a summary with a valid length.

3.3 Feature, score, and selection

Now, we give the definition of each of the features used in SumCR for sentence scoring and also present how to select sentences based on the final scores.

3.3.1 Exemplar: a subtopic-based feature

Based on the fuzzy memberships $u_{ci} \in U_{K\times N}$ and prototype weights $v_{cj} \in V_{K\times N}$ produced by PFC, two possible ways might be used to calculate the *Exemplar* weight, the one based on truncated clusters and the one based on original fuzzy clusters.

For the truncated case, we assign each sentence i to a cluster with the largest membership, i.e.,

$$x_i \in \mathcal{C}_k, \quad \text{with} \quad k = \arg\max_f u_{fi} \tag{8}$$

and we obtain the prototype weight of this sentence in the associated cluster $Pw_i = v_{ki}$. After that, we calculate the sum of similarities between this sentence to all other sentences in the same cluster, i.e.,

$$Sm_i = \sum_{j \neq i, j \in \mathcal{C}_k} s_{ij} \tag{9}$$

Table 2 Comparison of subtopic weights with non-subtopic weights

	T_{11}	T_{12}	T_{21}	T_{22}	T_{23}	T_{31}	T_{32}
Exemplar	0.3370	0.9104	0.4850	1.0000	0.3370	0.8079	1.0000
global	0.2308	0.9231	0.3077	0.7692	0.2308	0.8462	1.0000

where sentence j is in the same cluster $\mathcal{C}_k$ as sentence i. For a sentence i, both Sm_i and Pw_i reflect how central this sentence is in the related cluster. We hybridize these two measures to get *Exemplar* as below

$$Exemplar_i = (1 - \alpha) * Pw_i + \alpha * Sm_i \tag{10}$$

where α is the hybrid parameter. To make effective use of both Pw and Sm, α is set in (0, 1).

By truncating, some detailed information of the partition may be lost. To maintain those information, the *Exemplar* score can be calculated based on fuzzy assignments U as follows:

$$Smf_i(c) = \sum_j u_{ci} u_{cj} s_{ij} \tag{11}$$

$$Pwf_i(c) = u_{ci} v_{ci} \tag{12}$$

$$Exemplar_i = \max_c \{(1 - \alpha) * Smf_i(c) + \alpha * Pwf_i(c)\} \tag{13}$$

Normally, the results obtained with two strategies are close. We use the fuzzy one in the experiments of this paper.

For the previous example, based on the clustering results given in Table 1, we get the $Exemplar$ weight of each sentence in Table 2. To make comparison, we also computed the *global* score based on the whole sentence set, i.e., $global_i = \sum_{j \neq i} s_{ij} / \max_h \left(\sum_{j \neq h} s_{hj} \right)$, where s_{ij} is the similarity between two sentences in the whole collection. By comparison, we can see that the highest $Exemplar$ weights are assigned to the most representative sentences in both subtopics namely T_{22} and T_{32}; while the three largest $global$ weights are all assigned to sentences in the second topic. This illustrates that the subtopic-level $Exemplar$ has a better ability than corpus-level feature to provide diversity in content of the summary.

3.3.2 Position

We have just shown that the feature *Exemplar* plays a critical role in capturing subtopic-based representativeness. However, the effectiveness of this feature depends on the existence of the subtopic structure in the dataset. In other words, when the dataset does not contain obvious subtopics, such a cluster-based feature may become less effective. Therefore, to maintain a good performance on different datasets and further improve the quality of the summary produced, we also consider another feature, *Position*, which has been shown to be a good indication of the significance of a sentence. Early investigation on sentence position was made in [6], which shows that a very large portion of topic sentence comes as the first sentence of the paragraph. [10] found sentence location is the best individual feature compared with other three word-level features. Sentence position is also used as a default feature in MEAD. Here, we adopt a simple weighting scheme for *Position*, i.e., the score of *Position* decreases as the sentence becomes father from the beginning of the document. Specifically, we use Eq. (14) to assign the *Position* score to each sentence:

$$Position_i = \frac{1}{\sqrt{i}} \tag{14}$$

where i is the sentence index in the document it appears.

3.3.3 Score for generic summarization

After both scores of *Exemplar* and *Position* being calculated, we combine them to obtain the overall *Significance (Sig)* score of each sentence as

$$Sig_i = w_e * Exemplar_i + w_p * Position_i \tag{15}$$

where $w_e, w_p \geq 0$ are the combination parameters. The weights of each feature are typically scaled into [0, 1] before combination.

3.3.4 Score for task-focused summarization

The SumCR that has been presented so far is for generic summarization. To extend it for task-focused summarization, we need to further consider the information contained in the user-specified query when extracting sentences. A simple and widely used way is to consider the similarity between each sentence i and the query denoted as $SimToQ_i$ as a feature, and add it into the feature set. In such a way, our task-focused significance score $SigQ_i$ of each sentence i is weighted as below

$$SigQ_i = w_e * Exemplar_i + w_p * Position_i + w_s * SimToQ_i \tag{16}$$

where $w_e, w_p, w_s \geq 0$ are the respective combination parameter of each feature.

3.3.5 Selection criterion

Although the redundancy problem is expected to be alleviated by using subtopic-based feature with the clustering technique, a simple reranker may still be helpful in the final selecting of sentences to form a summary. We use the default reranker in MEAD in our experiment, which sorts the sentences by Sig (or $SigQ$) score computed in Eq. (15) [or Eq. (16)] in a descendent order, and successively decides whether to add the next sentence into the summary. The sentence is added to the summary if the current summary length does not exceed the required limitation and the similarities between the candidate sentence and the previous selected ones are all below a given threshold, e.g., 0.5 as used our experiment.

4 Experimental results

In this section, we carry out experimental studies to evaluate the proposed approach compared with other existing summarization systems. Both generic and task-focused multi-document summarization are conducted on DUC (Document Understanding Conferences)[1] datasets. Our experimental study mainly focus on two aspects, namely the effectiveness of *Exemplar* and the positive contribution of *Position* in SumCR.

[1] In the last few years, DUC http://duc.nist.gov/ has been established as a system evaluation competition for researchers to compare the performance of different summarization approaches on common datasets.

Table 3 Summarization systems

System ID	Description
SumCR-G, SumCR-Q	Proposed approach
MEAD [33]	A centroid-based approach
SNMF [37]	A clustering-based approach
HybHSum [8]	A probability model-based approach
DrS-G, DrS-Q [38]	A graph-based approach
HIERSUM [14]	A hierarchical LAD-style model based approach
Human-letter	The worst human performance provided by DUC
System-number	Top 3 automatic systems in DUC competition
Baseline	The baseline system used in DUC

4.1 Systems evaluated

We first compare the proposed SumCR with two most related approaches MEAD and SNMF. Two important features used by MEAD are *Centroid* and *Position*. Compared with subtopic-level feature *Exemplar* used in SumCR, which captures different aspects of the same topic and can be calculated based on sentence similarities measured with any advanced techniques, *Centroid* in MEAD does not capture subtopics and is defined based on the "bag of words" sentence representation. Another approach SNMF only relies on cluster-based feature to rank sentences. More comparison is given between SumCR and several other systems, such as supervised HybHSum, model-based HIERSUM, graph-based DrS (-G and -Q), and Top 3 DUC participated systems. The lowest-ranked human expert and baselines provided by DUC are also compared. A description of each system can be found in Table 3.

Here, for SumCR, we simply use $K = 3$ for all the topics of DUC2004 and DUC2006, and $K = 4$ for all the topics of DUC2005. We set $\alpha = 0.4$ [parameter in Eq. (10)] in all the experiments. The results of SNMF, HybHSum, HIERSUM, and DrS-G and DrS-Q are quoted from the corresponding original studies, where confidence internal is not given. For HybHSum and HIERSUM, only the results on DUC2006 have been reported in the original papers. The results of SNMF are reported in [37] on DUC2005 and DUC2006 where two types of sentence similarity are tested, i.e., the bag-of-words representation based (keyword) and semantic analysis based (SLSS). We implemented SNMF on DUC2004 to get its results on this dataset with keyword-based similarity as we used in SumCR. For SumCR and DrS, "-G" refers the version for generic summarization and "-Q" for query-focused summarization.

4.2 Datasets and preprocessing

Document Understanding Conferences (DUC) have been a forum for researchers in text summarization since 2001. Different tasks are assigned every year for researchers taking part into the DUC competition. The performance of different approaches for the same task are compared on the same benchmark datasets with common evaluation metrics. We use DUC2004 for generic summarization and DUC2005, DUC2006 for task-focused summarization. A summary of the tasks and datasets is given in Table 4. For each topic, documents are segmented into sentences and each narration section in the user profile in DUC2005 and DUC2006 is treated as a single query sentence. Since the focus of this paper is not on sentence similarity

Table 4 A summary of the tasks and datasets

	Generic	Task-focused	
	DUC2004	DUC2005	DUC2006
Number of topics	50	50	50
Number of documents of each topic	10	25–50	25
Data source	TDT	TREC	AQUAINT
Summary length	665 bytes	250 words	250 words

measure, we use the simple vector space model to present each sentence as a vector and each word as a feature. Cosine coefficient based on tf-isf (term frequency-inverse sentence frequency) weighting [29] is calculated as the pairwise sentence similarity. The indexing process is done by *Rainbow* [24] with default settings, i.e., stop-words removing but no stemming and feature selection.

4.3 Evaluation metric

The quality of a summary is evaluated with the ROUGE toolkit developed by [21], which is adopted by DUC for automatic summarization evaluation. ROUGE measures summary quality by counting the overlapping units such as n-gram, word sequences, and word pairs between the candidate summary and a reference summary (summaries). The former is often referred as an automatically generated summary, while the latter is normally produced by human experts. ROUGE-N is an n-gram recall measure computed as

$$\mathrm{ROUGE_N} = \frac{\sum_{S\in\{ref\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{ref\}}\sum_{gram_n\in S} Count(gram_n)} \tag{17}$$

where n represents the length of the n-gram, $Count_{match}(gram_n)$ stands for the maximum number of n-grams co-occurring in a candidate summary, and a set of reference summaries, and $Count(gram_n)$ is the number of n-gram in the reference summaries. Here, we report the mean value as well as 95% confidence interval over all topics of the recall scores of ROUGE-2 and ROUGE-SU4 (skip-bigram plus unigram) [20].

4.4 Comparison of results

We first compare SumCR with MEAD and SNMF. Table 5 shows the ROUGE scores of three approaches on three datasets and Table 6 gives the improvements of SumCR with respect to SNMF and MEAD, respectively on each dataset. From these two tables, it can be seen that SumCR performs consistently much better than the other two approaches for both generic and task-focused summarizations. Since both SumCR and MEAD use the *Position* feature, the improvement of SumCR compared with MEAD is attributed to *Exemplar*. Specifically, this indicates that the subtopic-level feature *Exemplar* in SumCR possibly works better than *Centroid* in MEAD in improving the summary quality. With the same keyword-based similarity, the results of clustering-based SNMF are much worse than those of SumCR. The performance of SNMF is improved considerably when semantic similarity is used instead of the keyword-based one.

Other than these two approaches, we compare SumCR with more other approaches in Table 7. For each year, there are over 30 systems have participated in DUC competition, and here we only compare with the top 3 systems. From this table, it is observed that although

Table 5 Comparison of SumCR with MEAD and SNMF

System ID	ROUGE-2	ROUGE-SU4
DUC2004		
SumCR-G	0.0965 [0.0871–0.1059]	0.1364 [0.1278–0.1453]
MEAD	0.0930 [0.0852–0.1011]	0.1319 [0.1253–0.1384]
SNMF (keyword)	0.0840 [0.0756–0.0916]	0.1266 [0.1187–0.1342]
DUC2005		
SumCR-Q	0.0700 [0.0620–0.0780]	0.1251 [0.1170–0.1330]
MEAD	0.0688 [0.0624–0.0795]	0.1232 [0.1155–0.1310]
SNMF (SLSS)	0.0604 [–]	0.1230 [–]
SNMF (keyword)	0.0571 [–]	0.1145 [–]
DUC2006		
SumCR-Q	0.0906 [0.0827–0.0988]	0.1437 [0.1364–0.1514]
SNMF (SLSS)	0.0855 [–]	0.1398 [–]
SNMF (keyword)	0.0830 [–]	0.1319 [–]
MEAD	0.0732 [0.0657–0.0808]	0.1235 [0.1159–0.1313]

Table 6 Improvement of SumCR with respect to MEAD and SNMF

	ROUGE-2 (%)	ROUGE-SU4 (%)
DUC2004		
MEAD	3.76	3.41
SNMF (keyword)	14.88	7.74
DUC2005		
MEAD	1.74	1.54
SNMF (SLSS)	15.89	1.71
SNMF (keyword)	22.59	9.26
DUC2006		
MEAD	23.77	16.36
SNMF (SLSS)	5.96	2.79
SNMF (keyword)	9.16	8.95

based on a simple similarity measure, SumCR-G achieves quite good results compared with those state-of-the-art approaches. SumCR-G produces the best result on DUC2004 for generic summarization, where the improvements of SumCR-G with respect to other approaches are significant. In task-focused summarization on DUC2005 and DUC2006, the performance of SumCR-Q ranks at 4 or 5th among all compared automatic summarizers. It can be seen that the results of SumCR-Q on DUC2005 are comparable to the 3rd best system participated in DUC2005. On DUC2006, the result of SumCR-Q is slightly better than DsR-Q and HIERSUM and is close to the result of HybHSum, which needs to be trained by manually generated summaries.

Here, all the results of SumCR-Q are produced based on a simple similarity measure, and the query information is only incorporated in a naive way. The keyword-based similarity used in SumCR-Q fails to capture the true similarity between two sentences which consist of different words but convey similar meanings; therefore, many approaches such as System-15

Table 7 More comparisons

System ID	ROUGE-2	ROUGE-SU4
DUC2004		
Human-G	0.0854 [0.0730–0.0979]	0.1291 [0.1188–0.1398]
SumCR-G	0.0965 [0.0871–0.1059]	0.1364 [0.1278–0.1453]
System-65	0.0920 [0.0830–0.1000]	0.1331 [0.1256–0.1404]
DsR-Q	0.0872 [0.0780–0.0960]	0.1290 [0.1209–0.1364]
System-104	0.0857 [0.0782–0.0934]	0.1294 [0.1231–0.1361]
System-35	0.0835 [0.0741–0.0932]	0.1286 [0.1215–0.1362]
Baseline	0.0640 [0.0555–0.0732]	0.1029 [0.0961–0.1101]
DUC2005		
Human-H	0.0887 [0.0780–0.1011]	0.1487 [0.1378–0.1606]
DsR-Q	0.0771 [0.0734–0.0808]	0.1337 [0.1303–0.1373]
System-15	0.0727 [0.0657–0.0795]	0.1318 [0.1247–0.1392]
System-17	0.0719 [0.0632–0.0805]	0.1299 [0.1214–0.1383]
SumCR-Q	0.0700 [0.0620–0.0780]	0.1251 [0.1170–0.1330]
System-10	0.0699 [0.0623–0.0776]	0.1253 [0.1186–0.1322]
Baseline	0.0402 [0.0330–0.0481]	0.0870 [0.0769–0.0971]
DUC2006		
Human-A	0.1032 [0.0911–0.1151]	0.1679 [0.1601–0.1764]
System-24	0.0957 [0.0877–0.1040]	0.1552 [0.1473–0.1628]
System-15	0.0912 [0.0835–0.0994]	0.1474 [0.1403–0.1547]
HybHSum	0.0910 [–]	0.1510 [–]
SumCR-Q	0.0906 [0.0827–0.0988]	0.1437 [0.1364–0.1514]
System-12	0.0899 [0.0819–0.0978]	0.1475 [0.1396–0.1549]
DsR-Q	0.0899 [0.0857–0.0943]	0.1427 [0.1391–0.1464]
HIERSUM	0.0860 [–]	0.1430 [–]
Baseline	0.0495 [0.0419–0.0575]	0.0979 [0.0896–0.1064]

and System-17 in DUC2005 use semantic similarity. The effectiveness of semantic similarity over keyword-based similarity also has been illustrated through the results of SNMF in Table 5. In task-focused summarization, the way of query information extraction from the task description directly affects the final summary. In System-15 participated in DUC2006, query terms are selected with a particular functional words list and stop-word list. Since a query may only contain a small number of words, query expansion [39] is one way to reserve and enrich the information contained in the original query. The results of SumCR-Q shown here are obtained without applying any query expansion or any term selection. We believe that SumCR-Q has the potential to achieve further improvements in its performance on task-focused summarization by incorporating sophisticated methods to make use of the query information in a more effective way.

4.5 Results with different w_p

To see how two features contribute to the final results, Tables 8, 9, and 10 show the mean values of ROUGE scores of SumCR on three datasets by combining *Exemplar* with *Position*

Table 8 Results of SumCR-G on DUC2004 with $w_e = 1$ and various w_p

w_p	ROUGE-2	ROUGE-SU4
0.0	0.0802	0.1250
0.2	0.0856	0.1272
0.4	0.0900	0.1315
0.6	0.0917	0.1334
0.8	0.0942	0.1343
1.0	0.0965	0.1364
SNMF	0.0840	0.1266

Table 9 Results of SumCR-Q on DUC2005 with $w_e = w_s = 1$ and various w_p

w_p	ROUGE-2	ROUGE-SU4
0.0	0.0668	0.1213
0.2	0.0681	0.1226
0.4	0.0700	0.1251
0.6	0.0696	0.1260
0.8	0.0693	0.1240
1.0	0.0695	0.1231
SNMF	0.0571	0.1145

Table 10 Results of SumCR-Q on DUC2006 with $w_e = w_s = 1$ and various w_p

w_p	ROUGE-2	ROUGE-SU4
0.0	0.0884	0.1419
0.2	0.0906	0.1437
0.4	0.0857	0.1380
0.6	0.0826	0.1342
0.8	0.0788	0.1300
1.0	0.0769	0.1273
SNMF	0.0830	0.1319

to various degrees. The ROUGE scores of keyword-based SNMF on each of the datasets are also listed in these three tables for the convenience of comparison.

When $w_p = 0$, the *Position* feature is totally not considered in SumCR. Comparing the results of SumCR with this setting on three datasets with those of SNMF, it shows that SumCR performs much better than SNMF on DUC2005 and DUC2006 and slightly worse on DUC2004. The difference in the performance of SumCR with $w_p = 0$ and SNMF indicates that the subtopic-level feature *Exemplar* of SumCR is more effective than the cluster-based feature used in SNMF.

From Table 8, it shows that on DUC2004, gradually increasing the combination weight of *Position* with w_p from 0 to 1 always leads to improvement in the results. For DUC2005 as in Table 9, the subtopic feature alone is not effective enough. Improvement is achieved by incorporating *Position*, and the best result comes with w_p around [0.4, 0.6]. Further emphasis on *Position* with $w_p > 0.6$ does not help to improve the overall performance. However, for DUC2006, it can be seen from Table 10 that $Exemplar$ alone is quite effective but improvement still can be achieved by combing with *Position* to a proper extent, e.g., $w_p = 0.2$. These results show that scoring sentences by taking into consideration of different types of features, e.g., *Exemplar* and *Position*, usually gives a more consistent good performance. However, even each individual feature is well defined, finding a proper combination weight for each feature is still not trivial. More efforts are needed in the future to work toward a sophisticated solution for this problem.

5 Conclusion and future work

We have proposed a new extractive approach SumCR for multi-document summarization where clustering is used to discover subtopics. Compared with other clustering-based approaches, SumCR provides a new subtopic feature *Exemplar*, which is shown to be more effective than other subtopic-level features; Other than *Exemplar*, we incorporated the document-level feature *Position*, which enables SumCR to work consistently better than approaches using only cluster-based features. Moreover, SumCR takes pairwise sentence similarities as input, which provides a convenient interface for making use of any advanced techniques of sentence similarity measure, while many existing approaches only accept the basic "a bag of words" representation.

Among over 30 DUC participated systems and several other state-of-the-art summarization systems, our method gives the best result in generic summarization on DUC2004 and ranks 4th or 5th on DUC2005 and DUC2006 for task-focused summarization. We also see the great potential of SumCR-Q from several aspects, for example: (1) in query-based summarization, SumCR-Q has not incorporated query expansion, query term selection or any other query processing techniques; (2) instead of using a more sophisticated similarity, SumCR-Q at moment only uses keyword-based similarity measure. We, therefore, believe that in the future, the performance of SumCR-Q possibly would be further improved by incorporating some advanced query processing techniques and sophisticated sentence similarity measures.

References

1. Aggarwal CC, Yu PS (2010) On clustering massive text and categorical data streams. Knowl Inf Syst 24(2):171–196
2. Aliguliyev RM (2006) A novel partitioning-based clustering method and generic document summarization. In: Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, pp 626–629
3. Amini MR, Gallinari P (2002) The use of unlabeled data to improve supervised learning for text summarization. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, pp 105–112
4. Arora R, Ravindran B (2008) Latent dirichlet allocation and singular value decomposition based multi-document summarization. In: Proceedings of the international conference on data mining, pp 713–718
5. Barzilay R, Lee L (2004) Catching the drift: probabilistic content models, with applications to generation and summarization. In: HLT-NAACL: proceedings of the main conference, 2004, pp 113–120
6. Baxendale P (1958) Machine-made index for technical literature-an experiment. IBM J Res Dev 2(4):354–361
7. Carbonell JG, Goldstein J (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, pp 335–336
8. Celikyilmaz A, Hakkani-Tur D (2010) A hybrid hierarchical model for multi-document summarization. In: Proceedings of the 48th annual meeting of the association for computational linguistics (ACL 2010), pp 1149–1154

9. Conroy JM, O'Leary DP (2001) Text summarization via hidden markov models. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, pp 406–407
10. Edmundson HP (1969) New methods in automatic extracting. J Assoc Comput Mach 16(2):264–285
11. Erkan G, Radev DR (2004) LexPageRank: prestige in multi-document text summarization. In: Proceedings of empirical methods in natural language (EMNLP 2004), pp 365–371
12. Feng S, Wang D, Yu G, Gao W, Wong K-F (2011) Extracting common emotions from blogs based on fine-grained sentiment clustering. Knowl Inf Syst 27(2):281–302
13. Gong Y, Liu X (2001) Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'01), pp 19–25
14. Haghighi A, Vanderwende L (2009) Exploring content models for multi-document summarization. In: Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the association for computational linguistics (NAACL'09), pp 362–370
15. Hovy E, Lin C-Y (1999) Automated text summarization in SUMMARIST. In: Mani I, Maybury M (eds) Advances in automatic text summarization. The MIT Press, Cambridge, pp 81–94
16. Jing H (2000) Sentence reduction for automatic text summarization. In: Proceedings of 6th conference on applied natural language processing (ANCL'00), pp 310–315
17. Knight K, Marcu D (2002) Summarization beyond sentence extraction: a probabilistic approach to sentence compression. Artif Intell 139(1):91–107
18. Kupiec J, Pedersen J, Chen F (1995) A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'95), pp 68–73
19. Lee J-H, Park S, Ahn C-M, Kim D (2009) Automatic generic document summarization based on non-negative matrix factorization. Inf Process Manag 45(1):20–34
20. Lin C-Y (2004) Rouge: a package for automatic evaluation of summaries. In: Text summarization branches out: proceedings of the ACL-04 workshop of ACL 2004, pp 74–81
21. Lin C-Y, Hovey E (2003) Automatic evaluation of summaries using n-gram co-occurence statistics. In: Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology, pp 71–78
22. Long C, Huang M, Zhu X, Li M (2009) Multi-document summarization by information distance. In: Proceedings of the 2009 Ninth IEEE international conference on data mining (ICDM'09), pp 866–871
23. Mani I (2001) Automatic summarization. John Benjamin's Publishing Company, Amsterdam
24. McCallum AK (1996) Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow/
25. Mei J-P, Chen L (2010) Fuzzy clustering with weighted medoids for relational data. Pattern Recognit 43(5):1964–1974
26. Mihalcea R, Tarau P (2004) TextRank: bringing order into texts. In: Dekang L, Dekai W (eds) Proceedings of empirical methods in natural language (EMNLP 2004), pp 404–411
27. Moschitti A (2009) Syntactic and semantic kernels for short text pair categorization. In: Proceedings of the 12th conference of the European chapter of the association for computational linguistics, Athens, pp 576–584
28. Nenkova A, Vanderwende L (2005) The impact of frequency on summarization. Technical Report, Microsoft Research, MSR-TR-2005-101
29. Neto JL, Santos AD, Kaestner CA, Freitas, AA (2000) Document clustering and text summarization. In: Proceedings of the 4th international conference on practical applications of knowledge discovery and data ming (PAKDD'00), pp 41–55
30. Nomoto T, Matsumoto Y (2001) A new approach to unsupervised text summarization. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'01), pp 26–34
31. Otterbacher JC, Radev DR, Luo A (2002) Revisions that improve cohesion in multi-document summaries: a preliminary study. In: Proceedings of the ACL02 workshop on automatic summarization, pp 27–36
32. Quan X, Liu G, Lu Z, Ni X, Wenyin L (2010) Short text similarity based on probabilistic topics. Knowl Inf Syst 25(3):473–491
33. Radev DR, Jing H, Stys M, Tam D (2004) Centroid-based summarization of multiple documents. Inf Process Manag 40(6):919–938
34. Shen D, Sun J-T, Li H, Yang Q, Chen, Z (2007) Document summarization using conditional random fields. In: Proceedings of the 20th international joint conference on artificial intelligence, pp 2862–2867
35. Tang J, Yao L, Chen D (2009) Multi-topic based query-oriented summarization. In: Proceedings of the SIAM international conference on data mining, pp 1147–1158

36. Vanderwende L, Suzuki H, Brockett C, Nenkova A (2007) Beyond sumbasic: task-focused summarization with sentence simplification and lexical expansion. Inf Process Manag 43(6):1606–1618
37. Wang D, Li T, Zhu S, Ding C (2008) Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'08), pp 307–314
38. Wei F, Li W, Lu Q, He Y (2010) A document-sensitive graph model for multi-document summarization. Knowl Inf Syst 22(2):245–259
39. Zhao L, Wu L, Huang X (2009) Using query expansion in graph-based approach for query-focused multidocument summarization. Inf Process Manag 45(1):35–41

Author Biographies

Jian-Ping Mei received her B.Eng. degree from School of Electronic and Information Engineering at Ningbo University, China, in 2005 and M.Eng. degree from School of Information Science and Electronic Engineering at Zhejiang University, China, in 2007. She is currently working toward Ph.D. degree at Nanyang Technological University in Singapore. Her research interests include clustering and applications to Web Documents.

Lihui Chen received the B.Eng. in Computer Science and Engineering at Zhejiang University, China and the Ph.D. in Computational Science at University of St. Andrews, UK. Currently, she is an Associate Professor in the Division of Information Engineering at Nanyang Technological University in Singapore. Her research interests include machine learning algorithms and applications, data mining, and web intelligence. She has published more than seventy referred papers in international journals and conferences in these areas. She is a senior member of the IEEE, and a member of the IEEE Computational Intelligence Society.