Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction

XIAOJUN WAN and JIANGUO XIAO
Peking University

Document summarization and keyphrase extraction are two related tasks in the IR and NLP fields, and both of them aim at extracting condensed representations from a single text document. Existing methods for single document summarization and keyphrase extraction usually make use of only the information contained in the specified document. This article proposes using a small number of nearest neighbor documents to improve document summarization and keyphrase extraction for the specified document, under the assumption that the neighbor documents could provide additional knowledge and more clues. The specified document is expanded to a small document set by adding a few neighbor documents close to the document, and the graph-based ranking algorithm is then applied on the expanded document set to make use of both the local information in the specified document and the global information in the neighbor documents. Experimental results on the Document Understanding Conference (DUC) benchmark datasets demonstrate the effectiveness and robustness of our proposed approaches. The cross-document sentence relationships in the expanded document set are validated to be beneficial to single document summarization, and the word cooccurrence relationships in the neighbor documents are validated to be very helpful to single document keyphrase extraction.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms: Algorithms, Experimentation

This article is an extended version of our conference papers presented at AAAI2007 and AAAI2008 [Wan and Yang 2007a; Wan and Xiao 2008a].
This research was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 60873155, the research Fund for the Doctoral Program of Higher Education (RFDP) under Grant No. 20070001059, Beijing Nova Program under Grant No. 2008B03, the National High Technology and Research and Development Program of China under Grant No. 2008AA01Z421, and the Program for New Century Excellent Talents in University (NCET) under Grant No. NCET-08-0006.
Authors' address: Institute of Computer Science and Technology, Peking University, Haidian District, Beijing 100817, China; email: {wanxiaojun, xiaojianguo}@icst.pku.edu.cn. X. Wan is also affiliated with Key Laboratory of Computational Linguistics (Peking University), MOE, Beijing 100871, China.

DOI 10.1145/1740592.1740596 http://doi.acm.org/10.1145/1740592.1740596

Additional Key Words and Phrases: Document summarization, keyphrase extraction, neighborhood knowledge, graph-based ranking

ACM Reference Format:
Wan, X. and Xiao, J. 2010. Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. ACM Trans. Inf. Syst. 28, 2, Article 8 (May 2010), 34 pages.
DOI = 10.1145/1740592.1740596 http://doi.acm.org/10.1145/1740592.1740596

1. INTRODUCTION

A summary is defined as a condensed version of the original document, and it usually consists of a few sentences that can highlight the major points of the document. A keyphrase is defined as a meaningful and significant expression consisting of one or more words in the document. The keyphrases of a document can be considered as a very short summary of the document.

The summary and keyphrases of a document can help readers get the gist of the original document in a short period of time. For example, readers always take a look at the abstract of an article before they read deep into it. Current search engines usually provide a short summary for each returned document and online news services usually produce a concise news summary to describe each news event. Moreover, document summaries and keyphrases can benefit many other text-related tasks, such as document retrieval, document clustering, and document categorization. Traditionally, the summary and keyphrases of a document are usually manually assigned by authors or experts. To date, a large number of scientific papers have author-assigned summaries and keyphrases. However, the vast majority of documents (e.g. news articles, magazine articles) on the Web do not have summaries and keyphrases, and it is hard to manually assign summaries and keyphrases for all the documents because it is time-consuming and costly. Therefore, it is necessary to develop automatic methods for document summarization and keyphrase extraction.

Document summaries can be classified into different types according to different dimensions. For example, a summary can be either *generic summary* or *query-relevant summary* (sometimes called *query-biased summaries*). A query-relevant summary is biased towards a given query or topic, and a generic summary is produced without any additional clues and prior knowledge. Furthermore, a summary can be either *abstraction-based* or *extraction-based*. An extraction-based summary involves merely selecting sentences or text segments from the source document, while an abstraction-based summary involves paraphrasing sections of the source document. In general, an abstraction-based summary can condense a text more strongly than an extraction-based summary, but it is harder than extraction-based summary because it requires the use of natural language generation technology. In this article, we focus on generic extraction-based summary. Similarly, keyphrases can be either *generic* or *query-relevant*, according to whether they are biased to a given query. Also, keyphrases can be either *assignation-based* or *extraction-based*. An assignation-based keyphrase may not appear in the source document, though it may appear in a predefined controlled vocabulary, while an extraction-based keyphrase appears in the source document, that is, the keyphrase is selected

from within the body of the source document. We focus on generic keyphrase extraction in this article.

Automatic document summarization and keyphrase extraction have drawn much attention for a long time. In the natural language processing and information retrieval communities, a series of workshops and conferences on automatic text summarization (e.g. NTCIR,[1] DUC[2]), special topic sessions in Association for Computational Linguistics (ACL), Conference on Computational Linguistics (COLING), and Special Interest Group on Information Retrieval (SIGIR) have advanced the summarization techniques and produced several experimental online systems. Note that document summarization and keyphrase extraction are domain-dependent tasks, for example, the characteristics of the summary and keyphrases for a scientific paper are different from the characteristics of the summary and keyphrases for a news article. This article focuses on document summarization and keyphrase extraction for news articles because news articles are one of the most popular document genres on the Web and they do not have associated summaries and keyphrases.

To the best of our knowledge, almost all previous methods have conducted the summarization and keyphrase extraction tasks by using only the information contained in the specified document, and the methods have extensively explored various syntactic and statistical features of a sentence or phrase in the source document, such as the position, the Part of Speech (POS) information, and the term weight. One common assumption of existing methods is that the documents are independent of each other and the extraction tasks are conducted separately without interactions for each document. However, a few topic-related documents actually have mutual influences with each other and they can provide useful clues to help extract summary or keyphrases from each other. For example, two documents sharing the topic "Hurricane Andrew" may present different aspects of the hurricane by using a portion of unique terms, but they usually share a few common terms to describe the central topic. Thus, the information presented in the summaries and the keyphrases of the documents are mostly related to each other; the two documents usually share a few common keyphrases (e.g. "Hurricane Andrew," "damage"), and the summaries of the two documents usually share much common information with each other. From a human's perspective, we would better understand a document if we have read more topic-related documents. Therefore, we believe that topic-related documents can provide additional knowledge for each other to better evaluate and extract genuine keyphrases and summaries from each other.

Based on this assumption, we propose constructing an appropriate knowledge context for a specified document by leveraging a few neighbor documents. The neighbor documents are then used to help extract the genuine summary and keyphrases from the specified document, because they can provide additional knowledge and more clues to evaluate the sentences and phrases in the specified document. In particular, the graph-based ranking algorithm is employed for single document summarization and keyphrase extraction by

[1]http://research.nii.ac.jp/ntcir/
[2]http://duc.nist.gov/

making use of the sentence-to-sentence and word-to-word relationships, respectively. Both the local information in the specified document and the global information in the neighbor documents are incorporated into the algorithm. For document summarization, two kinds of sentence-to-sentence relationships are used: the within-document relationships between sentences in the documents, and the cross-document relationships between sentences in different documents. For keyphrase extraction, two kinds of word-to-word relationships are used: the word cooccurrence relationships in the specified document, and the word cooccurrence relationships in the neighbor documents.

We have performed summarization experiments on the DUC2001 and DUC2002 (Document Understanding Conference) datasets with human-written summaries, and we have performed keyphrase extraction experiments on the DUC2001 datasets with human-annotated keyphrases. The experimental results demonstrate the effectiveness of the proposed approaches for both document summarization and keyphrase extraction. We find that the use of the cross-document relationships between sentences can improve the performance of single document summarization. And the use of the word relationships in the neighborhood context can significantly improve the performance of single document keyphrase extraction. We also investigate how the size of the neighborhood influences the summarization and keyphrase extraction performances. It is encouraging that a small number of neighbor documents can improve the performance.

The contributions of this paper are summarized as follows.

(1) We propose the novel idea of using neighborhood knowledge to improve single document summarization and keyphrase extraction for the first time.
(2) We implement the idea by incorporating the local information in the specified document and the global information in the neighbor documents into the graph-based ranking algorithm for document summarization and keyphrase extraction.
(3) We conduct experiments to validate the effectiveness of the proposed approaches on the DUC datasets, and thoroughly investigate how the parameters influence the final performance.

The rest of this article is organized as follows: Section 2 introduces related work. The proposed summarization and keyphrase extraction approaches are presented in Section 3. The experiments and results are given in Section 4. Last, we conclude our article in Section 5.

2. RELATED WORK

2.1 Document Summarization

As mentioned earlier, the methods for single document summarization can be either extraction-based or abstraction-based. Abstraction-based methods usually involve sentence reduction, compression and reformulation [Jing 2000; Jing and McKeown 2000; Knight and Marcu 2002]. We focus on extraction-based methods in this article.

Extraction-based methods usually involve assigning a saliency score to each sentence and then ranking the sentences in the document. The score is computed based on a combination of statistical and linguistic features, including term frequency [Luhn 1958], sentence position [Hovy and Lin 1997], cue words [Edmundson 1969], stigma words [Edmundson 1969], topic signature [Lin and Hovy 2000], lexical chains [Barzilay and Elhadad 1997; Silber and McCoy 2000], and so on. Machine learning methods have also been employed to extract sentences, including unsupervised clustering methods [Nomoto and Matsumoto 2001] and supervised learning methods, such as classification-based methods [Kupiec et al. 1995; Amini and Gallinari 2002], HMM-based methods [Conroy and O'Leary 2001] and CRF-based methods [Shen et al. 2007]. Other methods include maximal marginal relevance (MMR) [Carbonell and Goldstein 1998], latent semantic analysis (LSA) and relevance measure [Gong and Liu 2001]. McDonald and Chen [2002] propose a method based on text segmentation, in which the method ranks the text segments instead of the sentences. In Zha [2002], the mutual reinforcement between phrases and sentences is employed to iteratively extract key phrases and sentences from a document. Furthermore, Wan et al. [2007] propose a unified approach to simultaneous keyphrase extraction and document summarization by making use of three kinds of mutual reinforcement relationships: sentence-to-sentence relationships, word-to-word relationships, and sentence-to-word relationships.

In recent years, the graph-based ranking methods, including TextRank [Mihalcea and Tarau 2004, 2005] and LexRank [Erkan and Radev 2004] have been proposed for document summarization. Similar to Google's PageRank algorithm [Page et al. 1998] or Kleinberg's HITS algorithm [Kleinberg 1999], these methods first build a graph based on the similarity relationships among the sentences in a document and then the importance of a sentence is determined by taking into account the global information on the graph recursively, rather than relying only on the local sentence-specific information. The basic idea underlying the graph-based ranking algorithm is that of voting or recommendation. When one sentence links to another, it casts a vote for that sentence. The larger the number of votes that are cast for a sentence, the higher the importance of the sentence. Moreover, the importance of the sentence casting the vote determines how important the vote itself is. The computation of sentence importance is usually based on a recursive form, which can be transformed into the problem of solving the principal eigenvector of the transition matrix.

All of these methods make use of only the information contained in the specified document. The use of neighbor documents to improve single document summarization has not yet been investigated.

Other related work includes multidocument summarization [Radev et al. 2004; Radev and McKeown 1998; Harabagiu and Lacatusu 2005], query-relevant document summarization [Daumé and Marcu 2006], scientific article summarization [Teufel and Moens 2002], email summarization [Carenini et al. 2007], Web page summarization [Sun et al. 2005], book summarization [Mihalcea and Ceylan 2007], and so on.

Document summaries can help users seek information from a large corpus or on the Web [McDonald and Chen 2006]. In addition, document summaries have

been successfully used in the following IR and NLP tasks: document indexing [Sakai and Jones 2001], document classification [Kolcz et al. 2001; Shen et al. 2004], document clustering [Wang et al. 2004], and relevance feedback [Lam-Adesina and Jones 2001].

2.2 Keyphrase Extraction

Keyphrase assignment and keyphrase extraction are two major approaches to keyphrase generation for documents. Keyphrase extraction methods select phrases present in the source document; they usually consist of a candidate identification stage and a selection stage. In contrast to keyphrase extraction, keyphrase assignment methods typically select phrases, which may not appear in the source document, from a known and fixed phrase set, usually derived from a controlled vocabulary or taxonomy. Keyphrase assignment is usually performed by using the multiclass text classification techniques [Pouliquen et al. 2003; Medelyan and Witten 2006]. We focus on keyphrase extraction in this study. Keyphrase (or keyword) extraction methods can also be categorized into either unsupervised or supervised.

Unsupervised methods usually involve assigning a saliency score to each candidate phrase by considering various features. Krulwich and Burkey [1996] use heuristics to extract keyphrases from a document. The heuristics are based on syntactic clues, such as the use of italics, the presence of phrases in section headers, and the use of acronyms. Barker and Cornacchia [2000] propose a simple system for choosing noun phrases from a document as keyphrases. Muñoz [1997] uses an unsupervised learning algorithm to discover two-word keyphrases. The algorithm is based on Adaptive Resonance Theory (ART) neural networks. Steier and Belew [1993] use mutual information statistics to discover two-word keyphrases. Tomokiyo and Hurst [2003] use pointwise KL-divergence between multiple language models for scoring both phraseness and informativeness of phrases. More recently, Mihalcea and Tarau [2004] propose the TextRank model to rank keywords based on cooccurrence links between words. Such algorithms make use of voting or recommendations between words to extract keyphrases.

Supervised machine learning algorithms usually involve classifying a candidate phrase as to whether or not it is a keyphrase. GenEx [Turney 2000] and Kea [Frank et al. 1999; Witten et al. 1999] are two typical systems, in which the most important features for classifying a candidate phrase are the frequency and location of the phrase in the document. More linguistic knowledge has been explored by Hulth [2003]. Statistical associations between keyphrases have been used to enhance the coherence of the extracted keyphrases [Turney 2003]. Song et al. [2003] present an information gain-based keyphrase extraction system called KPSpotter. Nguyen and Kan [2007] focus on keyphrase extraction in scientific publications by using new features that capture salient morphological phenomena found in scientific keyphrases.

All of these methods make use of only the information contained in the specified document. The use of neighbor documents to improve single document keyphrase extraction has not yet been investigated.

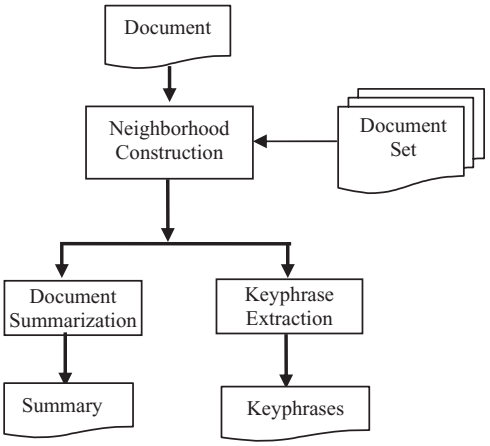

Fig. 1. The flowchart of the proposed approach.

Other related work includes terminology extraction [Park et al. 2002], Web page keyword extraction [Kelleher and Luz 2005] and finding advertising keywords [Yih et al. 2006].

Document keyphrases have been successfully used in the following IR and NLP tasks: document indexing [Gutwin et al. 1999], document classification [Krulwich and Burkey 1996], document clustering [Hammouda et al. 2005], and document summarization [Berger and Mittal 2000].

2.3 Collaborative Techniques

The idea underlying our work is related to collaborative techniques, which typically use the attributes of other similar objects to predict the attribute of the given object. In recent years, collaborative techniques have been successfully used in the following IR and NLP tasks: collaborative filtering [Xue et al. 2005], collaborative ranking [Wan and Yang 2007b; Wan and Xiao 2008b], collaborative recommendation [Balabanović and Shoham 1997] and Web mining [Wong et al. 2006]. The common idea is to make use of the interactions among multiple objects under the assumption that similar objects will have similar behaviors and characteristics.

3. THE PROPOSED APPROACH

3.1 Framework

We propose a unified framework to make use of neighborhood knowledge for both document summarization and keyphrase extraction. The framework consists of two steps: the first step is neighborhood construction and the second step is summary or keyphrase extraction using the neighborhood knowledge. Figure 1 gives the flowchart of the unified framework.

Given a specified document d_0 for summarization or keyphrase extraction, the proposed approach first finds a few neighbor documents for the document by document retrieval techniques. The neighbor documents are topically close to

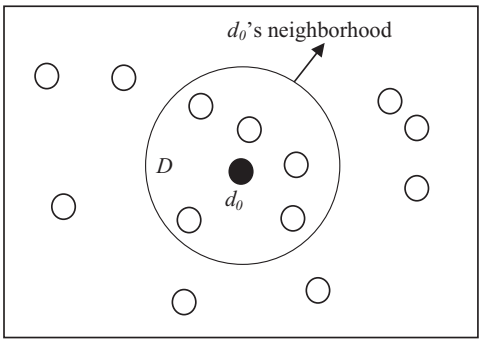

Fig. 2. Neighborhood construction for d_0.

the specified document and they construct the neighborhood knowledge context for the specified document. In other words, document d_0 is expanded to a small document set D, which provides more knowledge and clues for summary and keyphrase extraction from document d_0. Given the expanded document set, the proposed approach adopts the graph-based ranking algorithms for both document summarization and keyphrase extraction, though the implementation details of the algorithms are different. For document summarization, both the within-document sentence relationships (local information) and the cross-document sentence relationships (global information) within the context are incorporated into the summarization process. For keyphrase extraction, both the word co-occurrence relationships in the specified document (local information) and the word co-occurrence relationships in the neighbor documents (global information) are incorporated into the keyphrase evaluation process.

3.2 Neighborhood Construction

Given a specified document d_0, neighborhood construction aims to find a few nearest neighbors for the document from a text corpus or on the Web. The k neighbor documents $d_1, d_2, \ldots, d_k$ and the specified document d_0 construct the expanded document set $D = \{d_0, d_1, d_2, \ldots, d_k\}$ for d_0, which can be considered as the enlarged knowledge context for document d_0. The number k influences the summarization performance and will be investigated in the experiments. Figure 2 shows the neighborhood for document d_0.

The neighbor documents can be obtained by using the technique of nearest neighbor search. Nearest neighbor search aims to find a few neighbor documents in a text corpus, which are most similar to a given query document. The effectiveness of nearest neighbor search relies on the function for evaluating the similarity between two documents. In this study, we use the standard cosine measure for nearest neighbor search, similar to the document expansion technique used in Tao et al. [2006]. In the vector space model (VSM) [Baeza-Yates and Ribeiro-Neto 1999], a document d_i is represented by a vector $\vec{d_i}$ with each dimension referring to a unique term. The weight associated with term t is calculated by the $tf_{d_i,t} \times idf_t$ formula, where $tf_{d_i,t}$ is the number of occurrences of term t in document d_i and $idf_t = 1 + log(N/n_t)$ is the inverse document

frequency, where N is the total number of documents in the collection and n_t is the number of documents containing term t. The similarity $sim_{\text{doc}}(d_i, d_j)$, between documents d_i and d_j, can be defined as the normalized inner product of the two vectors $\vec{d}_i$ and $\vec{d}_j$:

$$sim_{\text{doc}}(d_i, d_j) = \frac{\vec{d}_i \cdot \vec{d}_j}{\|\vec{d}_i\| \times \|\vec{d}_j\|} \tag{1}$$

The efficiency of nearest neighbor search can be significantly improved by adopting some index structure in the implemented system, such as K-D-B tree, R-tree, SS-tree, SR-tree, or X-tree [Böhm and Berchtold 2001].

In the experiments, we simply use the cosine measure to compute the pairwise similarity value between the specified document d_0 and any document in the corpus, and then choose k documents (different from d_0) with the largest similarity values, as the nearest neighbors for d_0. Finally, there are a total of $k+1$ documents in the expanded document set. For the document set $D = \{d_0, d_1, d_2, ..., d_k\}$, the pairwise cosine similarity values between documents are calculated and recorded for later use.

The use of neighborhood information is worth more discussion. Because neighbor documents might not be sampled from the same generative model as the specified document, we probably do not want to trust them so much as the specified document [Tao et al. 2006]. Thus a confidence value is associated with every document in the expanded document set, which reflects our belief that the document is sampled from the same underlying model as the specified document. When a document is close to the specified one, the confidence value is high, but when it is farther apart, the confidence value will be reduced. Heuristically, we use the cosine similarity between a document and the specified document as the confidence value. The confidence values of the neighbor documents will be incorporated in the graph-based ranking algorithm.

3.3 Document Summarization

The proposed document summarization approach makes use of the constructed neighborhood knowledge by incorporating the cross-document relationships between sentences into the graph-based ranking algorithm. Like the PageRank algorithm [Page et al. 1998], the graph-based ranking algorithm employed in this study is essentially a way of deciding the importance of a sentence within a graph, based on global information recursively drawn from the entire graph. The basic idea is that of *voting* or *recommendation* between the sentences. A link between two sentences is considered as a vote cast from one sentence to the other sentence. The score associated with a sentence is determined by the votes that are cast for it, and the score of the sentences casting these votes.

The summarization algorithm is given in Figure 3.

In this algorithm, the first step aims to build a global affinity graph to reflect the relationships among all sentences in the expanded document set of $k+1$ documents. The second step aims to compute the informativeness score of each

Given document d_0 and the expanded document set $D=\{d_0, d_1, d_2, \ldots, d_k\}$, perform the following steps to extract the summary for d_0:

1. ***Neighborhood-Level Sentence Graph Building*** *Build a global affinity graph G based on all sentences of the documents in D; Let S denote the set of the sentences in the document set.*
2. ***Neighborhood-Level Sentence Evaluation*** *Based on the global affinity graph G, the graph-based ranking algorithm is employed to compute the informativeness score, IFScore(s_i), for each sentence s_i, where IFScore(s_i) quantifies the informativeness of the sentence s_i.*
3. ***Document-Level Redundancy Removing*** *A greedy algorithm is employed to remove redundancy for the informative sentences in d_0. Finally, the sentences of d_0 that are both informative and novel are chosen for the summary.*

Fig. 3. The algorithm for document summarization.

sentence based on the global affinity graph. The informativeness of a sentence indicates how much information about the main topic the sentence contains. The third step aims to remove redundant information in the summary and keep the sentences in the summary as novel as possible. A summary is expected to include the sentences with high informativeness and minimum redundancy. The details of the three steps are described in the next sections.

3.3.1 *Neighborhood-Level Sentence Graph Building.* Formally, given the expanded document set D, let $G = (S, E)$ be an undirected graph to reflect the relationships between sentences in the document set. $S = \{s_i | 1 \leq i \leq n\}$ is the set of sentences in the document set. E is the set of edges, which is a subset of $S \times S$. Each edge e_{ij} in E is associated with an affinity weight $sim_{sen}(s_i, s_j)$ between sentences s_i and s_j in S. The weight is calculated using the standard cosine measure as in Equation (1).

The links (edges) between sentences in the graph can be categorized into two classes: within-document link and cross-document link. Given a link between a sentence pair of s_i and s_j, if s_i and s_j come from the same document, the link is a within-document link; and if s_i and s_j come from different documents, the link is a cross-document link. Actually, the within-document links reflect the local information in a document, while the cross-document links reflect the global information in the expanded document set, which delivers mutual influences between documents in the set. The within-document links and the cross-document links are associated with different confidence values and the weight associated with each link is determined by both the corresponding sentence similarity value and the confidence value. Figure 4 demonstrates the two kinds of links between sentences.

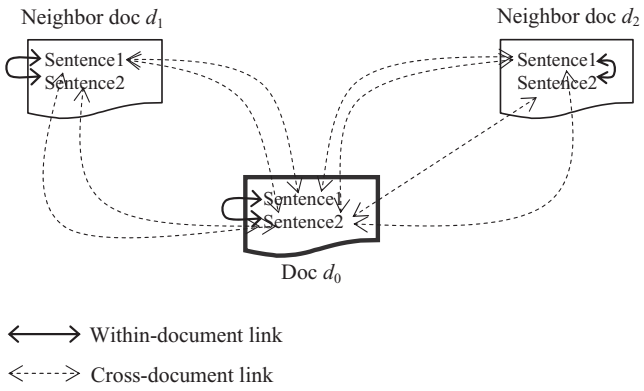

Fig. 4. Sample sentence links.

Graph G contains both kinds of links between sentences and is called the *Global Affinity Graph*. We use an adjacency (affinity) matrix M to describe G, with each entry corresponding to the weight of a link in the graph. $M = (M_{i,j})_{|S|\times|S|}$ is defined as follows:

$$M_{i,j} = \begin{cases} \lambda \times sim_{sen}(s_i, s_j), & \text{if } i \neq j, \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where λ specifies the confidence value of the sentence relationship. If the link between s_i and s_j is a within-document link, let $\lambda = 1$; if the link between s_i and s_j is a cross-document link, that is, s_i and s_j come from different documents d_k and d_l, let $\lambda = sim_{doc}(d_k, d_l)$.

Then M is normalized to $\tilde{M}$ as follows to make the sum of each row equal to 1[3]:

$$\tilde{M}_{i,j} = \begin{cases} M_{i,j} \Big/ \sum_{j=1}^{|S|} M_{i,j}, & \text{if } \sum_{j=1}^{|S|} M_{i,j} \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Similar to this process, another two affinity graphs G_{intra} and G_{inter} are also built: the within-document affinity graph G_{intra} includes only the within-document links between sentences (the entries of the cross-document links are set to 0); the cross-document affinity graph G_{inter} includes only the cross-document links between sentences (the entries of the within-document links are set to 0). The corresponding adjacency (affinity) matrices of G_{intra} and G_{inter} are denoted by M_{intra} and M_{inter}, respectively. M_{intra} and M_{inter} can be extracted from M and we have $M = M_{intra}+M_{inter}$. Similar to Equation (3), M_{intra} and M_{inter} are respectively normalized to $\tilde{M}_{intra}$ and $\tilde{M}_{inter}$ to make the sum of each row equal to 1.

3.3.2 *Neighborhood-Level Sentence Evaluation.* Based on the global affinity graph G, the informativeness score $IFScore_{all}(s_i)$ for sentence s_i can be

[3]The rows with all zero elements in $\tilde{M}$ are replaced by a smoothing vector with all elements set to $1/n$.

deduced from those of all other sentences linked with it and it can be formulated in a recursive form as follows:

$$IFScore_{all}(s_i) = \mu \cdot \sum_{all j \neq i} IFScore_{all}(s_j) \cdot \tilde{M}_{j,i} + \frac{(1-\mu)}{|S|}. \quad (4)$$

And the matrix form is:

$$\vec{\lambda} = \mu \tilde{M}^T \vec{\lambda} + \frac{(1-\mu)}{|S|}\vec{e}, \quad (5)$$

where $\vec{\lambda} = [IFScore_{all}(s_i)]_{|S|\times 1}$ is the vector of informativeness scores. $\vec{e}$ is a vector with all elements equal to 1. μ is the damping factor, usually set to 0.85 without further tuning. This process can be considered as a Markov chain by taking the sentences as the states. The corresponding transition matrix is given by $\mu\tilde{M}^T + \frac{(1-\mu)}{|S|}\vec{e}\vec{e}^T$. The stationary probability distribution of each state is obtained by the principal eigenvector of the transition matrix.

For implementation, the initial informativeness scores of all sentences are set to 1 and the iteration algorithm in Equation (4) is adopted to compute the new informativeness scores of the sentences. Usually the convergence of the iteration algorithm is achieved when the difference between the informativeness scores computed at two successive iterations for any sentences falls below a given threshold (0.0001 in this study).

Similarly, the informativeness score of sentence s_i can be deduced based on either the within-document affinity graph G_{intra} or the cross-document affinity graph G_{inter} as follows:

$$IFScore_{intra}(s_i) = \mu \cdot \sum_{all j \neq i} IFScore_{intra}(s_j) \cdot (\tilde{M}_{intra})_{j,i} + \frac{(1-\mu)}{|S|} \quad (6)$$

$$IFScore_{inter}(s_i) = \mu \cdot \sum_{all j \neq i} IFScore_{inter}(s_j) \cdot (\tilde{M}_{inter})_{j,i} + \frac{(1-\mu)}{|S|}. \quad (7)$$

The final informativeness score, $IFScore(s_i)$, of sentence s_i can be either $IFScore_{all}(s_i)$, $IFScore_{intra}(s_i)$, or $IFScore_{inter}(s_i)$. With different scenarios for computing the informativeness scores, three summarization methods are defined as follows:

UniformLink. $IFScore(s_i)$ is equal to $IFScore_{all}(s_i)$, that is, both the within-document relationships and the cross-document relationships are used.

InterLink. $IFScore(s_i)$ is equal to $IFScore_{inter}(s_i)$, that is, only the cross-document relationships are used.

IntraLink. $IFScore(s_i)$ is equal to $IFScore_{intra}(s_i)$, that is, only the within-document relationships are used.

We will investigate all of these summarization methods. Note that none of the previous graph-based ranking methods consider the cross-document links and they have $IFScore(s_i) = IFScore_{intra}(s_i)$, that is the IntraLink method is the baseline method.

3.3.3 *Document-Level Redundancy Removing*. This step aims to remove redundant information in the summary by penalizing sentences that are highly

1. *Initialize two sets A=Ø, B={s_i | i=1, 2, …, p}. Each sentence's affinity rank score is initialized to its information richness score, ARScore(s_i) = IFScore(s_i), i=1, 2, …, p.*
2. *Sort the sentences in B by their current affinity rank scores in descending order.*
3. *Suppose s_i is the highest ranked sentence, the first sentence in the ranked list. Move sentence s_i from B to A, and then impose the diversity penalty on the affinity rank score of each sentence linked with s_i as follows:*
   For each sentence s_j in B, we have
   $$ARScore(s_j) = ARScore(s_j) - (\tilde{M}_{d_0})_{j,i} \cdot IFScore(s_i).$$
4. *Go to step 2 and iterate until B= Ø or the iteration count reaches a predefined maximum number.*

Fig. 5. The algorithm for removing redundancy.

overlapping with other informative sentences. For the specified document d_0 to be summarized we can extract a subgraph G_{d_0} only containing the sentences in d_0 and the corresponding edges between them from the global affinity graph G. We assume document d_0 has p ($p<n$) sentences and the sentences' affinity matrix $M_{d_0} = (M_{d_0})_{p\times p}$ is derived from the original matrix M by extracting the corresponding entries. Then M_{d_0} is normalized to $\tilde{M}_{d_0}$ as Equation (3) to make the sum of each row equal to 1. The greedy algorithm in Zhang et al. [2005] is used to penalize the sentences highly overlapping with other informative sentences based on $\tilde{M}_{d_0}$. The algorithm is actually a variant form of the MMR algorithm and thus denoted as "MMR" in the next sections. The basic idea of the algorithm is to decrease the overall rank score of less informative sentences by the portion conveyed from the most informative one. The details of the algorithm are given in Figure 5.

Finally, the overall rank score for each sentence within the document is obtained. The sentences with highest overall rank scores are both highly informative and highly novel, and are chosen to be included in the summary for d_0 according to the summary length limit.

3.4 Keyphrase Extraction

The proposed keyphrase extraction approach makes use of the constructed neighborhood knowledge by incorporating the word cooccurrence relationships in the neighbor documents. The extraction algorithm is described in Figure 6.

In this framework, the first step aims to build a global affinity graph to reflect the neighborhood-level cooccurrence relationships between all candidate words in the expanded document set. The second step aims to compute the saliency scores of the candidate words by using the graph-based ranking algorithm. The saliency scores of the words indicate how much information about the main topic the words reflect. The third step aims to evaluate the candidate phrases in the specified document based on the neighborhood-level word scores, and

Given document d_0 and the expanded document set $D=\{d_0, d_1, d_2,\ldots, d_k\}$, perform the following steps to extract keyphrases for d_0:

1. ***Neighborhood-Level Word Graph Building*** *Build a global affinity graph G based on all candidate words restricted by syntactic filters in all the documents of the expanded document set D.*
2. ***Neighborhood-Level Word Evaluation*** *Employ the graph-based ranking algorithm to compute the global saliency score for each word.*
3. ***Document-Level Keyphrase Extraction*** *For the specified document d_0, evaluate the candidate phrases in the document based on the scores of the words contained in the phrases, and finally choose a few phrases with highest scores as the keyphrases of the document.*

Fig. 6. The algorithm for keyphrase extraction.

then choose a few salient phrases as the keyphrases of the document. This approach is called ExpandRank; the details of the three steps are described in the next sections.

3.4.1 *Neighborhood-Level Word Graph Building.* Formally, given the expanded document set D, let $G = (V, E)$ be an undirected graph to reflect the relationships between words in the document set. $V = \{v_i | 1 \leq i \leq l\}$ is the set of vertices and each vertex is a candidate word[4] in the document set. Because not all words in the documents are good indicators of keyphrases, the words added to the graph are restricted with syntactic filters, that is only the words with a certain part of speech are added. As in Mihalcea and Tarau [2004], the documents are tagged by a POS tagger, and only the nouns and adjectives are added into the vertex set.[5] E is the set of edges, which is a subset of $V \times V$. Each edge e_{ij} in E is associated with an affinity weight $aff(v_i,v_j)$ between words v_i and v_j. The weight is computed based on the cooccurrence relation between the two words, controlled by the distance between word occurrences. The cooccurrence relation can express cohesion relationships between words. Two vertices are connected if the corresponding words cooccur at least once within a window of maximum w words, where w can be set anywhere from 2 to 20 words. The affinity weight $aff(v_i,v_j)$ is simply set to be the count of the controlled cooccurrences between the words v_i and v_j in the whole document set as follows:

$$aff(v_i, v_j) = \sum_{d_p \in D} sim_{doc}(d_0, d_p) \times count_{d_p}(v_i, v_j), \tag{8}$$

where $count_{d_p}(v_i, v_j)$ is the count of the controlled cooccurrences between words v_i and v_j in document d_p, and $sim_{doc}(d_0, d_p)$ is the similarity factor, which reflects the confidence value for using document d_p $(0 \leq p \leq k)$ in the expanded document set.

[4]The original words are used without stemming.

[5]The corresponding POS tags of the candidate words include "JJ", "NN", "NNS", "NNP", "NNPS". We used the Stanford log-linear POS tagger [Toutanova and Manning 2000] in this study.

The graph is built based on the whole document set and it can reflect the global information in the neighborhood, which is called *Global Affinity Graph*. We use an affinity matrix M to describe G, with each entry corresponding to the weight of an edge in the graph. $M = (M_{i,j})_{|V|\times|V|}$ is defined as follows:

$$M_{i,j} = \begin{cases} aff(v_i, v_j), & \text{if } v_i \text{ links with } v_j \text{ and } i \neq j; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Then M is normalized to $\tilde{M}$ as follows to make the sum of each row equal to 1:

$$\tilde{M}_{i,j} = \begin{cases} M_{i,j} \Big/ \sum_{j=1}^{|V|} M_{i,j}, & \text{if } \sum_{j=1}^{|V|} M_{i,j} \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

3.4.2 *Neighborhood-Level Word Evaluation.* Based on the global affinity graph G, the saliency score $WordScore(v_i)$ for word v_i can be deduced from those of all other words linked with it and it can be formulated in a recursive form as in the PageRank algorithm:

$$WordScore(v_i) = \mu \cdot \sum_{all j \neq i} WordScore(v_j) \cdot \tilde{M}_{j,i} + \frac{(1-\mu)}{|V|}. \tag{11}$$

The matrix form is:

$$\vec{\lambda} = \mu \tilde{M}^T \vec{\lambda} + \frac{(1-\mu)}{|V|}\vec{e}, \tag{12}$$

where $\vec{\lambda} = [WordScore(v_i)]_{|V|\times 1}$ is the vector of word saliency scores. $\vec{e}$ is a vector with all elements equal to 1. μ is the damping factor, usually set to 0.85, as in the PageRank algorithm.

Likewise, this process can be considered as a Markov chain by taking the words as the states. The corresponding transition matrix is given by $\mu\tilde{M}^T + \frac{(1-\mu)}{|V|}\vec{e}\vec{e}^T$. The stationary probability distribution of each state is obtained by the principal eigenvector of the transition matrix. For implementation, the initial scores of all words are set to 1 and the iteration algorithm in Equation (11) is adopted to compute the new scores of the words. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any words falls below a given threshold (0.0001 in this study).

3.4.3 *Document-Level Keyphrase Extraction.* After the scores of all candidate words in the document set have been computed, candidate phrases (either single-word or multiword) are selected and evaluated for the specified document d_0. The candidate words (nouns and adjectives) of d_0, which is a subset of V, are marked in the text of document d_0, and sequences of adjacent candidate words are collapsed into a multiword phrase. The phrases ending with an adjective are not allowed; only the phrases ending with a noun are collected as candidate phrases for the document. For instance, in the following sentence: "*Mad/JJ cow/NN disease/NN has/VBZ killed/VBN 10,000/CD cattle/NNS,*" the candidate phrases are "*Mad cow disease*" and "*cattle*." The score of a

Table I. Summary of Datasets

	DUC 2001	DUC 2002
Task	Task 1	Task 1
Number of documents	309	567
Number of clusters	30	59
Data source	TREC-9	TREC-9
Summary length	100 words	100 words

candidate phrase p_i is computed by summing the neighborhood-level saliency scores of the words contained in the phrase.

$$PhraseScore(p_i) = \sum_{v_j \in p_i} WordScore(v_j) \tag{13}$$

All the candidate phrases in document d_0 are ranked in decreasing order of the phrase scores and the top m phrases are selected as the keyphrases of d_0. m ranges from 1 to 20 in this study.

4. EXPERIMENTS AND RESULTS

4.1 Document Summarization

4.1.1 *Evaluation Setup.* The experiments were performed on Task 1 of DUC2001 [Over 2001] and Task 1 of DUC2002 [Over and Liggett 2002]. The two tasks aimed to evaluate generic summaries with a length of approximately 100 words or less. DUC2001 provided 309 English news articles collected from TREC-9, and DUC2002 provided 567 English news articles collected from TREC-9 for the single-document summarization task. The DUC2001 documents could be categorized into 30 news topics and the DUC2002 documents could be categorized into 59 news topics, which guaranteed that we could find topic-related documents for a specified document. Table I gave a short summary of the two datasets. The sentences in each article have been separated and the sentence information has been stored files. Each dataset was considered as the corpus for document expansion for any specified document in the dataset, which could be easily expanded by adding more documents. Each specified document was expanded by adding k documents (different from the specified document) most similar to the document. For the similarity calculation between sentences or documents, the stopwords were removed and the remaining words were stemmed using Porter's stemmer [Porter 1980].

We used the widely used ROUGE toolkit [Lin and Hovy 2003] for automatic summarization evaluation. It measured summary quality by counting overlapping units such as the n-gram, word sequences, and word pairs between the candidate summary and the reference summary. ROUGE-N was an n-gram recall measure computed as follows:

$$ROUGE-N = \frac{\sum_{S \in \{RefSum\}} \sum_{n\text{-}gram \in S} Count_{match}(n-gram)}{\sum_{S \in \{RefSum\}} \sum_{n\text{-}gram \in S} Count(n-gram)}, \tag{14}$$

Table II. Comparison Results on DUC2001 (w/ MMR)

System	ROUGE-1	ROUGE-2	ROUGE-W
UniformLink(k = 1)	0.43531 [0.42761 – 0.44455]	0.15836	0.13589
UniformLink(k = 5)	0.43745 [0.43175 – 0.44559]	0.16033	0.13599
UniformLink(k = 10)	**0.43786** [0.42989 – 0.44673]	**0.16138**	**0.13635**
InterLink(k = 1)	0.43069 [0.42133 – 0.43894]	0.15743	0.13459
InterLink(k = 5)	0.43570 [0.42716 – 0.44374]	0.15965	0.13555
InterLink(k = 10)	0.43673 [0.42760 – 0.44503]	0.15985	0.13571
IntraLink	0.43335 [0.42373 – 0.44144]	0.15568	0.13566

Table III. Comparison Results on DUC2001 (w/o MMR)

System	ROUGE-1	ROUGE-2	ROUGE-W
UniformLink(k = 1)	0.42962 [0.42043 – 0.43953]	0.15878	0.13392
UniformLink(k = 5)	**0.43451** [0.42612 – 0.44610]	**0.16194**	**0.13574**
UniformLink(k = 10)	0.43356 [0.42543 – 0.44425]	0.16179	0.13528
InterLink(k = 1)	0.42684 [0.41555 – 0.43503]	0.15750	0.13353
InterLink(k = 5)	0.43417 [0.42588 – 0.44411]	0.16134	0.13569
InterLink(k = 10)	0.43105 [0.42122 – 0.44100]	0.16009	0.13461
IntraLink	0.42941 [0.41998 – 0.43913]	0.15810	0.13483

where n stood for the length of the n-gram, and $Count_{match}(n\text{-}gram)$ was the maximum number of n-grams cooccurring in a candidate summary and a set of reference summaries. $Count(n\text{-}gram)$ was the number of n-grams in the reference summaries.

The ROUGE toolkit reported separate scores for 1, 2, 3, and 4-gram, and also for longest common subsequence cooccurrences. Among these different scores, the unigram-based ROUGE score (ROUGE-1) has been shown to most agree with human judgment [Lin and Hovy 2003]. We showed three of the ROUGE metrics in the experimental results: ROUGE-1 (unigram-based), ROUGE-2 (bigram-based), and ROUGE-W (based on weighted longest common subsequence, weight = 1.2). We also showed the 95% confidence interval reported by the toolkit within square brackets for the ROUGE-1 score. Note that the significant tests in Wan and Yang [2007a] were performed by using t-test. In order to truncate summaries longer than the length limit, we used the "-l" option in the ROUGE toolkit.

4.1.2 *Performance Comparison.* The proposed approach considering neighbor documents (UniformLink) is compared with the baseline method relying only on the specified document (IntraLink). We also show the results of InterLink to demonstrate how reliable the cross-document relationships are. Tables II and IV show the comparison results after removing redundancy ("w/ MMR") on the DUC2001 and DUC2002 datasets, respectively. Tables III and V show the comparison results before removing redundancy ("w/o MMR") on the two datasets, respectively. For the methods of UniformLink and InterLink, the parameter k is heuristically set to 1, 5, and 10, respectively.

Table IV. Comparison Results on DUC2002 (w/ MMR)

System	ROUGE-1	ROUGE-2	ROUGE-W
UniformLink(k = 1)	0.46562 [0.45631 – 0.47484]	0.19557	0.16056
UniformLink(k = 5)	0.46738 [0.45952 – 0.47602]	0.19618	0.16156
UniformLink(k = 10)	**0.47162** [0.46198 – 0.48209]	**0.20114**	**0.16314**
InterLink(k = 1)	0.46641 [0.45808 – 0.47423]	0.19430	0.16060
InterLink(k = 5)	0.46703 [0.45799 – 0.47683]	0.19574	0.16141
InterLink(k = 10)	0.46870 [0.45898 – 0.47701]	0.19800	0.16211
IntraLink	0.46261 [0.45414 – 0.46863]	0.19457	0.16018

Table V. Comparison Results on DUC2002 (w/o MMR)

System	ROUGE-1	ROUGE-2	ROUGE-W
UniformLink(k = 1)	0.46034 [0.45101 – 0.47022]	0.19543	0.15966
UniformLink(k = 5)	0.46000 [0.45121 – 0.47105]	0.19478	0.15907
UniformLink(k = 10)	0.46360 [0.45489 – 0.47482]	0.19777	0.16068
InterLink(k = 1)	0.45925 [0.45036 – 0.46887]	0.19433	0.15861
InterLink(k = 5)	**0.46396** [0.45576 – 0.47215]	**0.19813**	**0.16084**
InterLink(k = 10)	0.46345 [0.45387 – 0.47421]	0.19701	0.16075
IntraLink	0.45591[0.44620 – 0.46308]	0.19201	0.15789

As can be seen from the tables, the proposed UniformLink always outperforms the baseline IntraLink on both datasets, no matter whether the process of removing redundancy is applied, which shows that the neighborhood knowledge does benefit single document summarization. Moreover, we can see that the InterLink method can perform as well as the baseline IntraLink on the DUC2001 dataset, and it can even outperform IntraLink on the DUC2002 dataset, which demonstrates that the cross-document relationships between sentences in the expanded document set are reliable enough to evaluate and extract salient sentences from a single document. Actually, the expanded document set is about the same topic as the specified document and the important information contained in the specified document would be also contained in other documents, although perhaps in different representations. Thus the knowledge from the neighbor documents would help to analyze and extract important information from the specified document.

4.1.3 *Influences of Neighbor Document Number k.* In order to investigate how the size of the expanded document set influences the summarization performance, we conducted experiments with different values of k. Figures 7–9 show the ROUGE-1, ROUGE-2, and ROUGE-W values for different methods with different values of k on the DUC2001 dataset, respectively. Figures 10–12 show the performance values for different methods with different values of k on the DUC2002 dataset, respectively. In the figures, k ranges from 1 to 15, indicating there are totally 2 to 16 documents in the expanded document sets. Four methods are investigated, including UniformLink and InterLink, with and without the process of removing redundancy ("w/ MMR" and "w/o MMR").

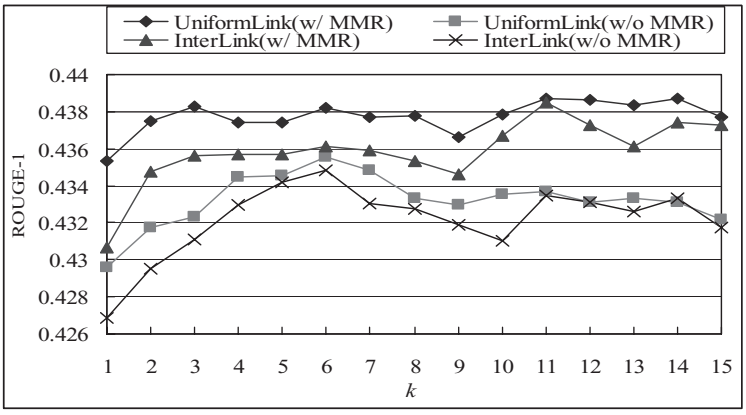

Fig. 7. ROUGE-1 performance vs. k on DUC2001.

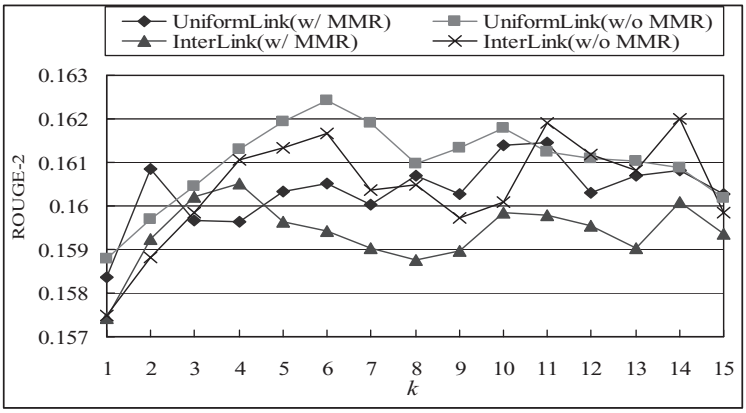

Fig. 8. ROUGE-2 performance vs. k on DUC2001.

Seen from the figures, the summarization performance does not always increase with k: the performances tend to decrease or at least stop increasing when k is large. The performance curves show that the use of more neighbor documents is unnecessary, and will even deteriorate the performance because having more neighbor documents runs a risk of introducing more noise. Thus the size of the expanded set can be set to a small number, which will improve the computational efficiency and make the proposed approach more applicable.

4.1.4 *Relative Contributions of Within-Document and Cross-Document Links.* In order to investigate the relative contributions of the within-document links and cross-document links to the summarization performance, we set a parameter to distinguish the two kinds of links in the affinity graph. The new affinity matrix is defined as follows:

$$M = \lambda \cdot M_{intra} + (1 - \lambda) \cdot M_{inter}, \tag{15}$$

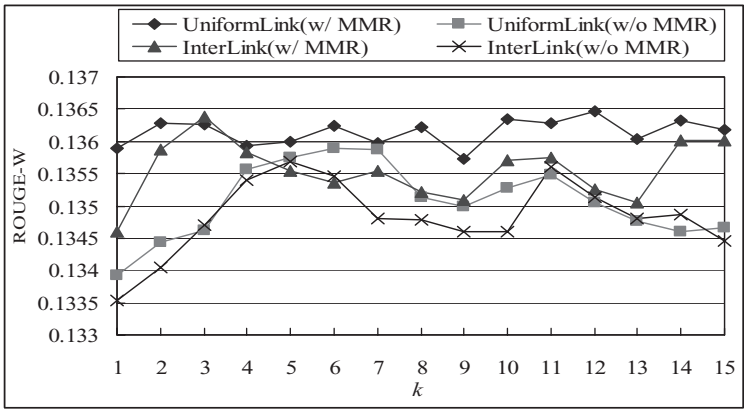

Fig. 9. ROUGE-W performance vs. k on DUC2001.

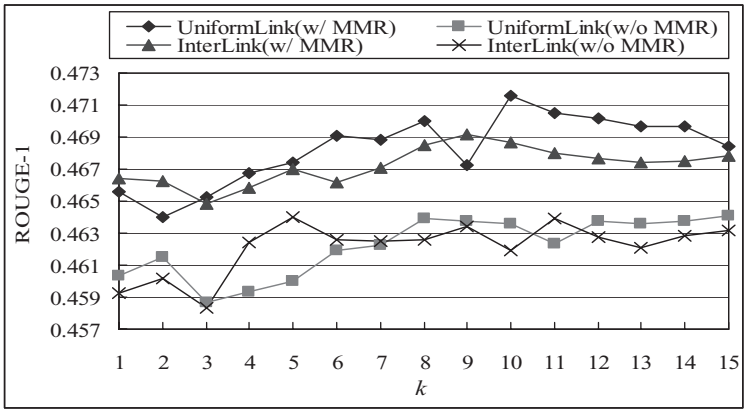

Fig. 10. ROUGE-1 performance vs. k on DUC2002.

where $\lambda \in [0,1]$. The informativeness scores of the sentences are then computed based on the new matrix. Figures 13–16 show the summarization performance curves (ROUGE-1 and ROUGE-W) with respect to different λ on the DUC2001 and DUC2002 datasets, respectively. Here, the neighbor document number k is fixed at 10. As can be seen from the figures, both kinds of sentence links can contribute to the summarization performance, but the cross-document links are more important than the within-document links.

4.1.5 *Comparison with DUC Systems.* Our proposed UniformLink method ($k = 10$) is compared with five typical participating systems on DUC2002.[6] The typical systems are the systems with high ROUGE scores, chosen from the participating systems on the single document summarization task of DUC2002. Table VI gives a short summary of the five systems. Table VII shows the

[6]The summarization results for participating systems on DUC2001 are incomplete, thus we only compare our proposed system with the DUC2002 participating systems.

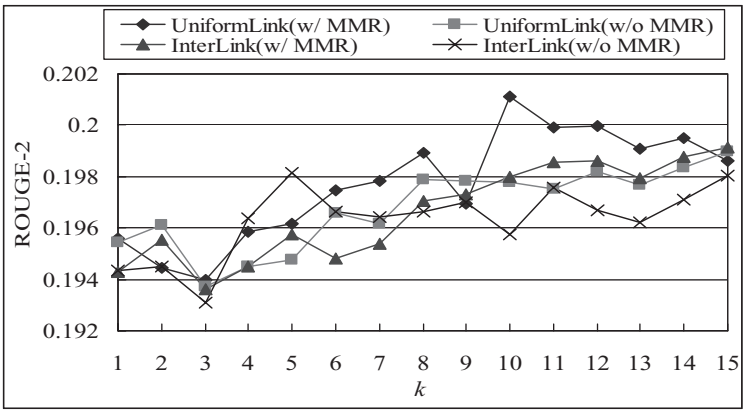

Fig. 11. ROUGE-2 performance vs. k on DUC2002.

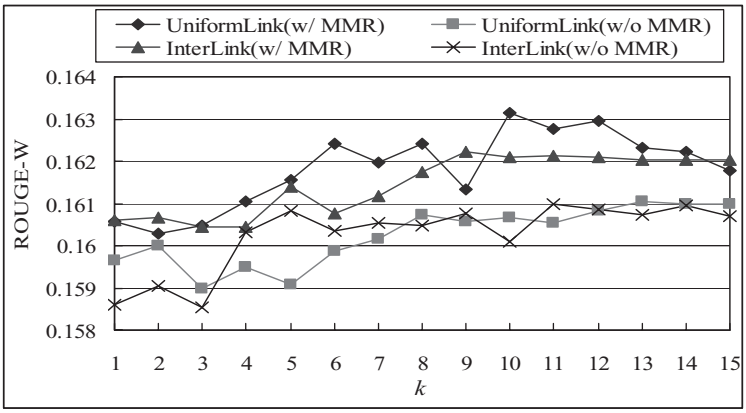

Fig. 12. ROUGE-W performance vs. k on DUC2002.

comparison results. In the tables, we use the system codes to represent the typical systems. As can be seen from the tables, the performance of our proposed method is comparable with that of the best participating systems.

4.2 Keyphrase Extraction

4.2.1 *Evaluation Setup.* To our knowledge, there was no gold standard news dataset with assigned keyphrases for evaluation. So we manually annotated the DUC2001 dataset [Over 2001] and used the annotated dataset for evaluation in this study. The dataset was originally used for document summarization. As mentioned earlier, the dataset consisted of 309 news articles collected from TREC-9, in which two articles were duplicate (d05a\FBIS-41815 and d05a\FBIS-41815∼), so the actual number of the documents used in the experiments is 308. The articles could be categorized into 30 news topics and the average length of the documents was 740 words. Two graduate students were employed to manually label the keyphrases for each document. At most

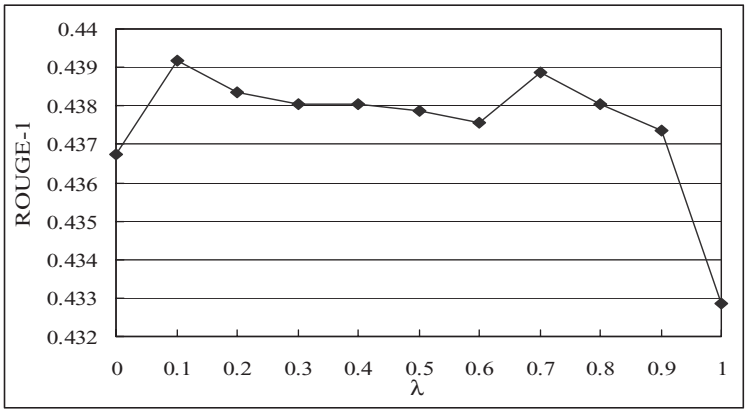

Fig. 13. ROUGE-1 performance vs. λ on DUC2001.

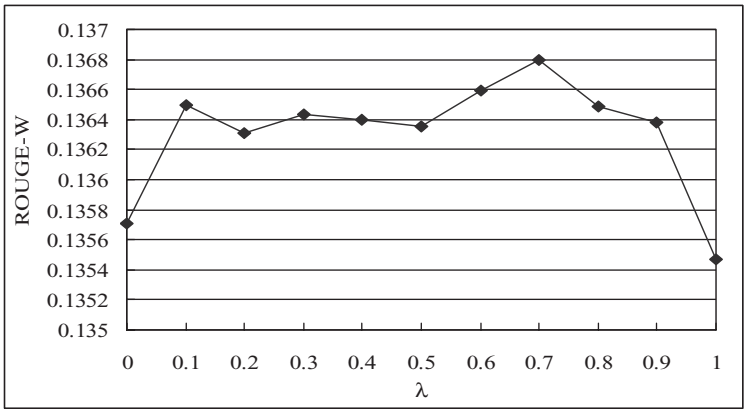

Fig. 14. ROUGE-W performance vs. λ on DUC2001.

10 keyphrases could be assigned to each document. The annotation process lasted two weeks. The Kappa statistic for measuring interagreement among annotators was 0.70. The annotation conflicts between the two subjects were solved by discussion. Finally, 2488 keyphrases were labeled for the dataset. The average keyphrase number per document was 8.08 and the average word number per keyphrase was 2.09. In the experiments, the DUC2001 dataset was considered as the corpus for document expansion in this study, which could be easily expanded by adding more documents. Each specified document was expanded by adding k documents (different from the specified document) most similar to the document.

For evaluation of keyphrase extraction results, the automatic extracted keyphrases were compared with the manually labeled keyphrases. The words in a keyphrase were converted to their corresponding basic forms using word stemming before comparison. We used the popular precision, recall, and

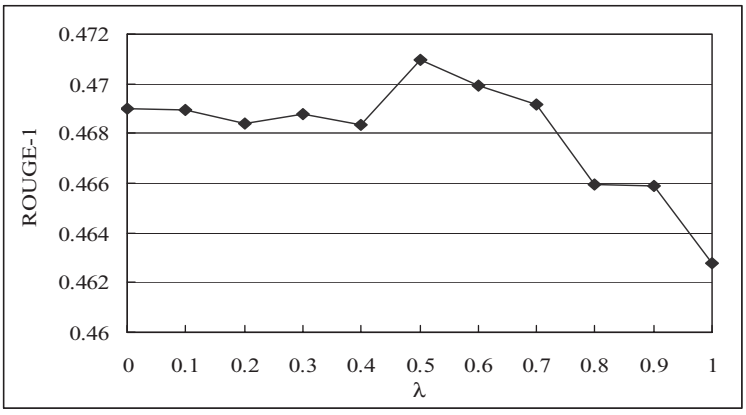

Fig. 15. ROUGE-1 performance vs. λ on DUC2002.

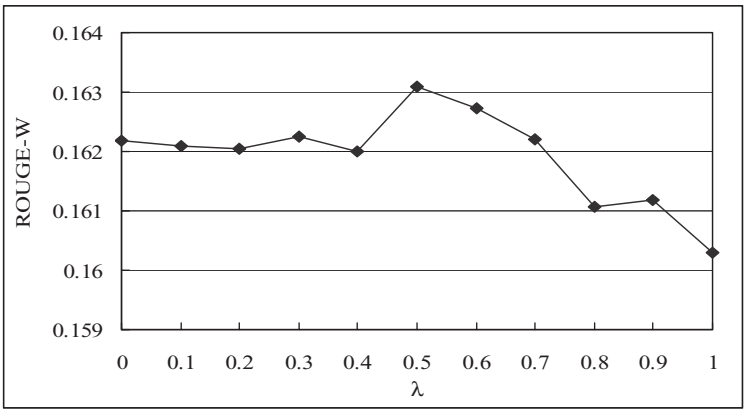

Fig. 16. ROUGE-W performance vs. λ on DUC2002.

F-measure as evaluation metrics; they were defined as follows:

$$precision = \frac{count_{correct}}{count_{system}} \tag{16}$$

$$recall = \frac{count_{correct}}{count_{human}} \tag{17}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}, \tag{18}$$

where $count_{correct}$ was the total number of correct keyphrases extracted by the system, $count_{system}$ was the total number of automatically extracted keyphrases, and $count_{human}$ was the total number of human-labeled keyphrases.

4.2.2 *Performance Comparison.* The proposed approach (ExpandRank) is compared with the baseline methods relying only on the specified document

Table VI. Summary of Typical Participating Systems on DUC2002

System ID	System Code	Group	System Description
Ccsnsa.v2	28	CCS-NSA	Supervised sentence extraction with HMM and LRM
Wpdv-xtr.v1	21	Catholic Univ. Nijmegen	Supervised sentence classification with WPDV
ULeth 131m	31	Univ. of Lethbridge	Unsupervised sentence extraction + text segmentation
Kul.2002	29	Catholic Univ. Leuven	Unsupervised sentence extraction + topic segmentation
Ntt.duc02	27	NTT	Supervised sentence classification with SVM

Table VII. System Comparison Results on DUC2002

System	ROUGE-1	ROUGE-2	ROUGE-W
28	0.48049	0.22832	0.17073
21	0.47754	0.22273	0.16814
UniformLink(k = 10)	0.47162	0.20114	0.16314
31	0.46506	0.20392	0.16162
29	0.46384	0.21246	0.16462
27	0.46019	0.21273	0.16342

(SingleRank and TFIDF). The SingleRank baseline uses the graph-based ranking algorithm to compute the word scores for each single document based on the local graph for the specified document. The TFIDF baseline computes the word scores for each single document based on the word's TFIDF value in the specified document. The two baselines do not make use of the neighborhood knowledge.

Table VIII gives the comparison results of the baseline methods and the proposed ExpandRank methods with different neighbor numbers ($k = 1, 5, 10$). In the experiments, the keyphrase number m is typically set to 10 because at most 10 keyphrases can be manually labeled for each document, and the cooccurrence window size w is also simply set to 10.

As can be seen from Table VIII, the ExpandRank methods with different neighbor numbers can always outperform the baseline methods of SingleRank and TFIDF over all three metrics. The results demonstrate the effectiveness of the proposed method.

Table VIII. Keyphrase Extraction Results

System	Precision	Recall	F-measure
TFIDF	0.232	0.281	0.254
SingleRank	0.247	0.303	0.272
ExpandRank(k = 1)	0.264	0.325	0.291
ExpandRank(k = 5)	**0.288**	**0.354**	**0.317**
ExpandRank(k = 10)	0.286	0.352	0.316

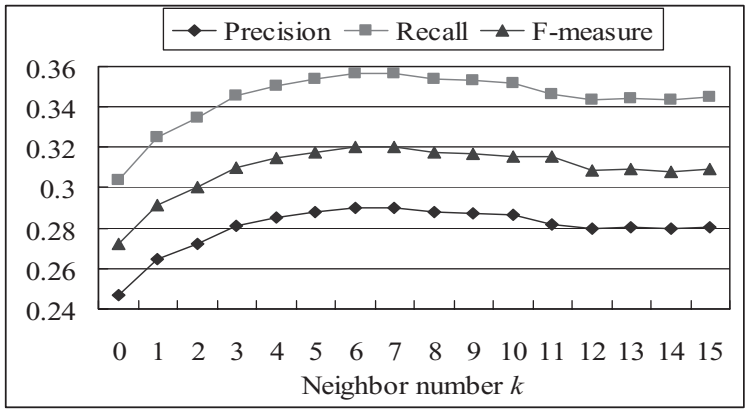

Fig. 17. ExpandRank ($m = 10$, $w = 10$) performance vs. neighbor number k.

4.2.3 *Influences of Neighbor Document Number k.* In order to investigate how the size of the neighborhood influences the keyphrase extraction performance, we conducted experiments with different values of the neighbor number k. Figure 17 shows the performance curves for the ExpandRank method. In the figure, k ranges from 0 to 15. Note that when $k = 0$, the ExpandRank method degenerates into the baseline SingleRank method. We can see from the figure that the performance of ExpandRank (i.e. $k>0$) can always outperform the baseline SingleRank method ($k = 0$), no matter how many neighbor documents are used. We can also see that the performance of ExpandRank first increases and then decreases with the increase of k. The trend demonstrates that very few or very many neighbors will deteriorate the results, because very few neighbors cannot provide sufficient knowledge and very many neighbors may introduce noisy knowledge. Seen from the figure, it is not necessary to use many neighbors for ExpandRank; the neighbor number can be set to a relatively small number (5), which will improve the computational efficiency and make the proposed approach more applicable.

4.2.4 *Influences of Window Size w.* In order to investigate how the cooccurrence window size influences the keyphrase extraction performance, we conducted experiments with different values of window size w. Figures 18 and 19 show the performance curves for ExpandRank when w ranges from 2 to 20. In Figure 18 the neighbor number is set to 5 and in Figure 19 the neighbor number is set to 10. We can see from the figures that the performance is almost not affected by the window size, except when w is set to 2.

4.2.5 *Influences of Keyphrase Number m.* In these experiments, the keyphrase number is set to 10. We further conducted experiments with different values of the keyphrase number m to investigate how the keyphrase number influences the keyphrase extraction performance. Figures 20 and 21

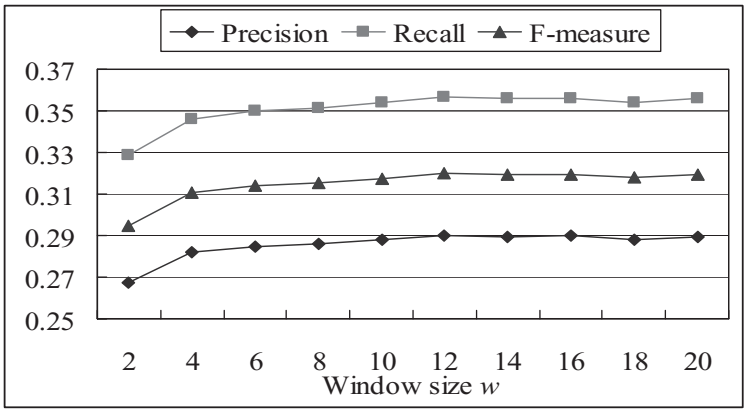

Fig. 18. ExpandRank ($k = 5$, $m = 10$) performance vs. window size w.

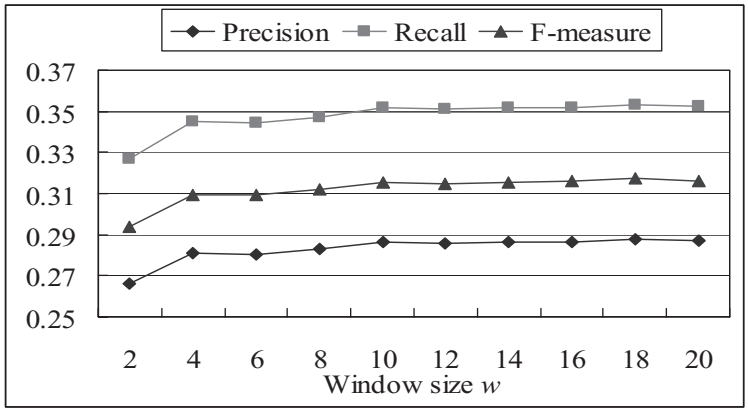

Fig. 19. ExpandRank ($k = 10$, $m = 10$) performance vs. window size w.

show the performance curves for ExpandRank when m ranges from 1 to 20. In Figure 20 the neighbor number is set to 5 and in Figure 21 the neighbor number is set to 10. We can see from the figures that the precision values decrease with the increase of m, and the recall values increase with the increase of m; while the F-measure values first increase and then tend to decrease with the increase of m.

4.2.6 *Relative Contributions of Single Document and Neighbor Documents.* In order to investigate the relative contributions from the single document and the neighbor documents to the final performance, we set a parameter to distinguish the two kinds of contributions in the computation of word affinity weight as follows:

$$aff(v_i, v_j) = \lambda \cdot count_{d_0}(v_i, v_j) + (1 - \lambda) \cdot \sum_{d_p \in D \wedge p \neq 0} sim_{doc}(d_0, d_p) \times count_{d_p}(v_i, v_j), \tag{19}$$

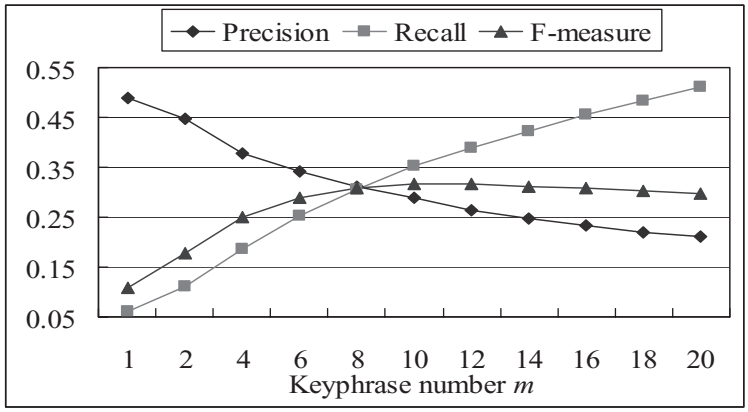

Fig. 20. ExpandRank ($k = 5$, $w = 10$) performance vs. keyphrase number m.

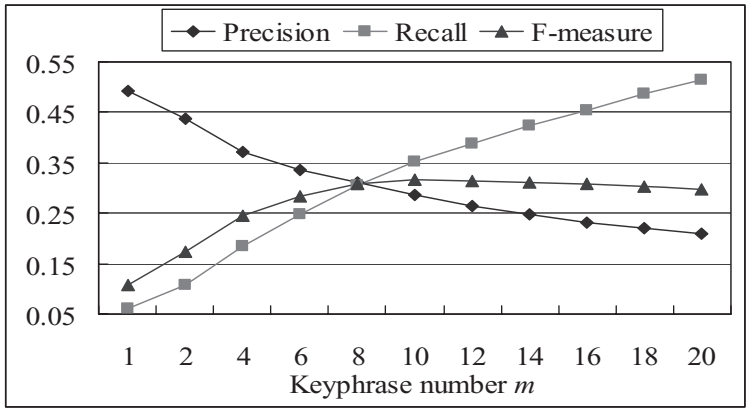

Fig. 21. ExpandRank ($k = 10$, $w = 10$) performance vs. keyphrase number m.

where $\lambda \in [0,1]$. The word scores are then computed based on the new affinity matrix. This fusion method is equal to SingleRank when λ is set to 1.

Figure 22 shows the extraction performance curves with respect to different λ. Here, other parameters k, w, and m are typically fixed to 10. As can be seen from the figures, both the word relationships in the single document and the word relationships in the neighbor documents are beneficial to the extraction performance, but the neighbor documents contribute more than the single document.

4.3 Discussion

The use of neighborhood knowledge will increase the computational complexity of the summarization and keyphrase extraction approaches, because of two reasons.

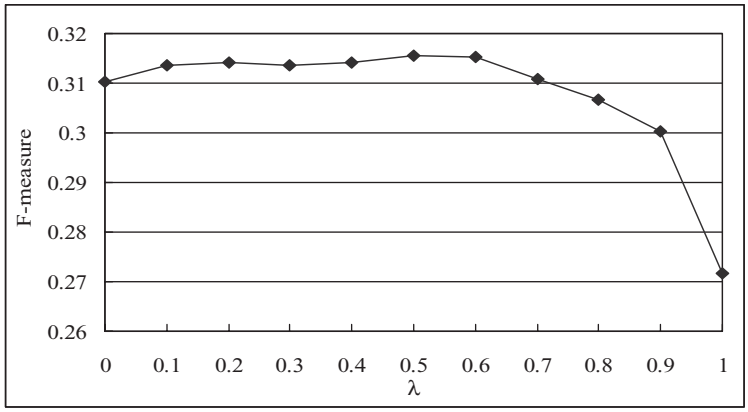

Fig. 22. Extraction Results (F-measure) vs. Fusion Weight λ ($k = 10$, $w = 10$, $m = 10$).

(1) The proposed approaches involve finding a few neighbor documents in a large corpus by using a search engine for nearest neighbor search. The computational complexity of this step relies on the implementation of the search engine. The worst case is that the query document is compared with each document in the corpus. The computational complexity can be improved to a constant by using advanced index structures.

(2) The proposed approaches involve processing the additional neighbor documents in the sentence or phrase ranking phase. In the proposed summarization approach, the number of the vertices in the sentence graph is $(k+1)\times Ave_{sentence}$ and the number of the edges is at most $((k+1)\times Ave_{sentence})^2/2$, where $Ave_{sentence}$ represents the average sentence number of the documents. In contrast, in the baseline IntraLink approach, the number of the vertices in the sentence graph is $Ave_{sentence}$ and the number of the edges is at most $Ave_{sentence}{}^2/2$. Likewise, in the proposed keyphrase extraction approach, the number of vertices in the word graph is $(k+1)\times Ave_{word}$ and the number of the edges is at most $((k+1)\times Ave_{word})^2/2$, where Ave_{word} represents the average sentence number of the documents. In contrast, in the baseline SingleRank approach, the number of vertices in the word graph is Ave_{word} and the number of edges is at most $Ave_{word}{}^2/2$.

The overall efficiency can be improved by collaboratively conducting single document summarization or keyphrase extraction in a batch mode. Suppose there are multiple documents for summarization and keyphrase extraction separately, we can group the documents into clusters, and for each cluster we can use all other documents as the neighbors for a specified document. Thus the mutual influences among all documents can be incorporated into the summarization or keyphrase extraction algorithm and all the sentences or words in the documents of a cluster are evaluated collaboratively, resulting in single summarizations and keyphrase extractions of all the documents in a batch mode.

Original Text for D05\AP900322-0200 on DUC2001

``Mad cow disease'' has killed 10,000 cattle, restricted the export market for Britain's cattle industry and raised fears about the safety of eating beef.

The government insists the disease poses only a remote risk to human health, but scientists still aren't certain what causes the disease or how it is transmitted.

``I think everyone agrees that the risks are low,'' says Martin Raff, a neurobiologist at University College, London. ``But they certainly are not zero. I have not changed my eating habits, but I certainly do wonder.''

Mad cow disease, or bovine spongiform encephalopathy, or BSE, was diagnosed only in 1986. The symptoms are very much like scrapie, a sheep disease which has been in Britain since the 1700s. The incurable disease eats holes in the brains of its victims; in late stages a sick animal may act skittish or stagger drunkenly.

The suspicion is that the disease was transmitted through cattle feed, which used to contain sheep by-products as a protein supplement.

The government banned the use of sheep offal in cattle feed in June 1988, and later banned the use of cattle brain, spleen, thymus, intestines and spinal cord in food for humans. Sheep offal is still used in pig and poultry feed.

Earlier this month, the government announced it would pay farmers 100 percent of market value or average market price, whichever is less, for each animal diagnosed with BSE.

``I think it is a recognition _ not just of pressure from farmers _ but that the public would feel more confident that no BSE-infected animal would ever be likely to go anywhere near the food chain if there was 100 percent compensation,'' said Sir Simon Gourlay, president of the National Farmers Union.

The disease struck one of his own cows, Gourlay said. ``In the course of 24 hours, the animal went from being ostensibly quite normal to very vicious and totally disoriented.''

As of Feb. 9, the Ministry of Agriculture, Fisheries and Food said that 9,998 cattle have been destroyed after being diagnosed with BSE.

The government has paid $6.1 million in compensation, and is budgeting $16 million for 1990.

Ireland's Department of Agriculture and Food said about 20 cases have been confirmed there, all of them near the border with the British province of Northern Ireland.

Because of the disease, the U.S. Department of Agriculture's Animal and Plant Health Inspection Service banned imports of cattle, embryos and bull semen from Great Britain in July, said Margaret Webb, a USDA spokeswoman in Washington.

Similar embargoes have been imposed by Australia, Finland, Israel, Sweden, West Germany and New Zealand, according to the agriculture ministry, and the European Community has proposed a ban on exports of British cattle older than 6 months.

David Maclean, a junior agriculture minister, has complained of ``BSE hysteria'' in the media and has insisted that the risk of the disease passing to humans is ``remote.''

The government has committed $19 million to finding the cause of the disease.

A commission chaired by Professor Sir Richard Southwood of Oxford University reported last year that the cause of BSE ``is quite unlike any bacteria or known viruses.''

The report said the disease was impossible to detect in apparently healthy animals because it did not prompt the immune system to produce antibodies.

The Southwood report said it was ``most unlikely'' that the disease was a threat to humans. But the report added: ``If our assessments of these likelihoods are incorrect, the implications would be extremely serious.''

There is a human variant of spongiform encephalopathy, known as Creutzfeldt-Jakob disease. About two dozen cases were reported in Britain last year.

Another form, known as kuru, had been found cannibals in New Guinea.

According to a report in the British Medical Journal, the incidence of Creutzfeldt-Jakob disease is no higher in Britain than it is in countries free of scrapie.

``It is urgent that the same reassurance can be given about the lack of effect of BSE on human health,'' a consultative committee reported to the agriculture ministry. The committee's report, released early this year, said it is only a ``shrewd guess'' that BSE was transmitted through sheep offal in cattle feed.

Fig. 23. Sample document.

Finally, we show the summaries and keyphrases extracted by our proposed methods and the baseline methods for a sample document in Figures 23–25. The sample document is given in Figure 23, and the extracted summaries and keyphrases are shown in Figures 24 and 25, respectively.

100 words summary extracted by the proposed UniformLink (k=10)

[1] Mad cow disease, or bovine spongiform encephalopathy, or BSE, was diagnosed only in 1986.
[2] The suspicion is that the disease was transmitted through cattle feed, which used to contain sheep by-products as a protein supplement.
[3] ``Mad cow disease'' has killed 10,000 cattle, restricted the export market for Britain's cattle industry and raised fears about the safety of eating beef.
[4] The government insists the disease poses only a remote risk to human health, but scientists still aren't certain what causes the disease or how it is transmitted.
[5] The symptoms are very much like scrapie, a sheep disease which has been in Britain since the 1700s.

100 words summary extracted by the baseline IntraLink

[1] Mad cow disease, or bovine spongiform encephalopathy, or BSE, was diagnosed only in 1986.
[2] The symptoms are very much like scrapie, a sheep disease which has been in Britain since the 1700s.
[3] The government insists the disease poses only a remote risk to human health, but scientists still aren't certain what causes the disease or how it is transmitted.
[4] As of Feb. 9 Fisheries and Food said that 9,998 cattle have been destroyed after being diagnosed with BSE.
[5] The suspicion is that the disease was transmitted through cattle feed, which used to contain sheep by-products as a protein supplement.

Fig. 24. Summaries extracted by UniformLink and IntraLink for the sample document.

10 keyphrases extracted by the proposed ExpandRank (k=10)

mad cow disease; sheep disease; creutzfeldt-jakob disease; incurable disease; disease passing; british cattle; cattle brain; cattle feed; cattle industry; british medical journal;

10 keyphrases extracted by the baseline SingleRank

sheep disease; mad cow disease; creutzfeldt-jakob disease; incurable disease; disease passing; british cattle; cattle feed; cattle industry; cattle brain; plant health inspection service;

Fig. 25. Keyphrases extracted by ExpandRank and SingleRank for the sample Document.

5. CONCLUSION AND FUTURE WORK

This article proposes a novel approach to single document summarization and keyphrase extraction by leveraging the neighborhood knowledge of the specified document. For document summarization, the within-document relationships and the cross-document relationships between sentences are incorporated in the graph-based ranking algorithm. The additional knowledge provided by the neighbor documents is acquired through the cross-document sentence relationships. For keyphrase extraction, the word cooccurrence relationships in the neighbor documents are incorporated in the algorithm. Experimental results on the DUC2001 and DUC2002 datasets demonstrate the effectiveness of the proposed summarization approach and the results also show the importance of the cross-document relationships between sentences. Experimental results on the DUC2001 datasets demonstrate the effectiveness of the proposed keyphrase extraction approach.

In this study, only the graph-based ranking algorithm is adopted for evaluating sentences or words in the documents. In future work, other sentence or word ranking algorithms will be integrated into the proposed framework to validate the robustness of the technique of leveraging neighbor documents. We will also integrate the traditional linguistic features of each sentence into the graph-based ranking algorithm. Moreover, we will make use of the rich link

information between Web pages to acquire additional knowledge for extracting summaries or keyphrases from Web pages.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable and constructive comments.

REFERENCES

AMINI, M. R. AND 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(SIGIR)*. 105–112.

BAEZA-YATES, R. AND RIBEIRO-NETO, B. 1999. *Modern Information Retrival*. ACM Press/Addison Wesley.

BALABANOVIĆ, M. AND SHOHAM, Y. 1997. Fab: content-based, collaborative recommendation. *Comm. ACM 40*, 3, 66–72.

BARKER, K. AND CORNACCHIA, N. 2000. Using nounphrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*. 40–52.

BARZILAY, R. AND ELHADAD, M. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*. 10–17.

BERGER, A. AND MITTAL, V. 2000. OCELOT: A system for summarizing Web Pages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and development in Information Retrieval (SIGIR)*. 144–151.

BÖHM, C. AND BERCHTOLD, S. 2001. Searching in high-dimensional spaces-index structures for improving the performance of multimedia databases. *ACM Comput. Surv. 33*, 3, 322–373.

CARBONELL, J. AND 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 (SIGIR)*. 335–336.

CARENINI, G., NG, R. T., AND ZHOU, X. 2007. Summarizing email conversations with clue words. In *Proceedings of the 16th International Conference on World Wide Web*. 91–100.

CONROY, J. M. AND O'LEARY, D. P. 2001. Text summarization via Hidden Markov Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 406–407.

DAUMÉ, H. AND MARCU, D. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*. 305–312.

EDMUNDSON, H. P. 1969. New methods in automatic abstracting. *J. ACM 16*, 2, 264–285.

ERKAN, G. AND RADEV, D. R. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. 22*, 457–479.

FRANK, E., PAYNTER, G. W., WITTEN, I. H., GUTWIN, C., AND NEVILL-MANNING, C. G. 1999. Domain-specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*. 668–673.

GONG, Y. H. AND 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)*. 19–25.

GUTWIN, C., PAYNTER, G. W., WITTEN, I. H., NEVILL-MANNING, C. G., AND FRANK, E. 1999. Improving browsing in digital libraries with keyphrase indexes. *J. Dec. Support Syst. 27*, 81–104.

HAMMOUDA, K. M., MATUTE, D. N., AND KAMEL, M. S. 2005. CorePhrase: keyphrase extraction for document clustering. In *Proceedings of IAPR 4th International Conference on Machine Learning and Data Mining (MLDM)*. 265–274.

HARABAGIU, S. AND LACATUSU, F. 2005. Topic themes for multidocument summarization. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 202–209.

HOVY, E. AND LIN, C. Y. 1997. Automated text summarization in SUMMARIST. In *Proceedings of ACL/EACL Worshop on Intelligent Scalable Text Summarization*. 18–24.

HULTH, A. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 216–223.

JING, H. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP)*. 310–315.

JING, H. AND MCKEOWN, K. R. 2000. Cut and paste based text summarization. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (NAACL)*. 178–185.

KELLEHER, D. AND LUZ, S. 2005. Automatic hypertext keyphrase detection. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. 1608–1609.

KLEINBERG, J. M. 1999. Authoritative sources in a hyperlinked environment. *J. ACM 46*, 5, 604–632.

KNIGHT, K. AND MARCU, D. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artif. Intell. 139*, 1, 91–107.

KOLCZ, A., PRABAKARMURTHI, V., AND KALITA, J. 2001. Summarization as feature selection for text categorization. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM)*. 365–370.

KRULWICH, B. AND BURKEY, C. 1996. Learning user information interests through the extraction of semantically significant phrases. In *Spring Symposium on Machine Learning in Information Access (AAAI)*. 110–112.

KUPIEC, J., PEDERSEN, J., AND 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)*. 68–73.

LAM-ADESINA, A. M. AND JONES, G. J. F. 2001. Applying summarization techniques for term selection in relevance feedback. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1–9.

LIN, C. Y. AND HOVY, E. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th Conference on Computational Linguistics (ACL)*. 495–501.

LIN, C. Y. AND HOVY, E. H. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*. 71–78.

LUHN, H. P. 1958. The automatic creation of literature abstracts. *IBM J. Res. Devel. 2*, 2, 159–165.

MCDONALD, D. AND CHEN, H. 2002. Using sentence-selection heuristics to rank text segment in TXTRACTOR. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*. 28–35.

MCDONALD, D. AND CHEN, H. 2006. Summary in context: searching versus browsing. *ACM Trans. Inform. Syst. 24*, 1, 111–141.

MEDELYAN, O. AND WITTEN, I. H. 2006. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*. 296–297.

MIHALCEA, R. AND TARAU, P. 2004. TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 404–411.

MIHALCEA, R. AND TARAU, P. 2005. A language independent algorithm for single and multiple document summarization. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP): Companion Volume including Posters/Demos and Tutorial Abstracts*. 19–24.

MIHALCEA, R. AND CEYLAN, H. 2007. Explorations in automatic book summarization. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*. 380–389.

MUÑOZ, A. 1997. Compound key word generation from document databases using a hierarchical clustering ART model. *Intell. Data Anal. 1*, 1–4, 25–48.

NGUYEN, T. D. AND KAN, M.-Y. 2007. Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries (ICADL)*. 317–326.

NOMOTO, T. AND 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),* 26–34.

OVER, P. 2001. Introduction to DUC-2001: an intrinsic evaluation of generic news text summarization systems. In *Proceedings of the DUC'01 Workshop on Text Summarization*.

OVER, P. AND LIGGETT, W. 2002. Introduction to DUC: an intrinsic evaluation of generic news text summarization systems. In *Proceedings of the DUC'02 Workshop on Text Summarization.*

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The pagerank citation ranking: Bringing order to the web. *Tech. Rep.*, Stanford Digital Libraries.

PARK, Y., BYRD, R. J., AND BOGURAEV, B. 2002. Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics*. 1–7.

PORTER, M. F. 1980. An algorithm for suffix stripping. *Program 14*, 3, 130–137.

POULIQUEN, B., STEINBERGER, R., AND IGNAT, C. 2003. Automatic annotation of multilingual text collections with a conceptual thesaurus. In *Proceedings of the Workshop 'Ontologies and Information Extraction' at the Summer School 'The Semantic Web and Language Technology - Its Potential and Practicalities' (EUROLAN)*. 9–28.

RADEV, D. R., JING, H. Y., STYS, M., AND TAM, D. 2004. Centroid-based summarization of multiple documents. *Inform. Proc. Manag. 40*, 6, 919–938.

RADEV, D. R. AND MCKEOWN, K. R. 1998. Generating natural language summaries from multiple on-line sources. *Comput. Ling. 24*, 3, 469–500.

SAKAI, T. AND JONES, K. S. 2001. Generic summaries for indexing in information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 190–198.

SHEN, D., CHEN, Z., YANG, Q., ZENG, H.-J., ZHANG, B., LU, Y., AND MA, W.-Y. 2004. Web-page classification through summarization. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR).* 242–249.

SHEN, D., SUN, J.-T., LI, H., YANG, Q., AND CHEN, Z. 2007. Document Summarization using conditional random fields. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. 2862–2867.

SILBER, H. G. AND MCCOY, K. 2000. Efficient text summarization using lexical chains. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*. 252–255.

SONG, M., SONG, I.-Y., AND HU, X. 2003. KPSpotter: a flexible information gain-based keyphrase extraction system. In *Proceedings of the 5th ACM International Workshop on Web Information and Data Management (WIDM)*, 50–53.

STEIER, A. M. AND BELEW, R. K. 1993. Exporting phrases: A statistical analysis of topical language. In *Proceedings of the Second Symposium on Document Analysis and Information Retrieval*. 179–190.

SUN, J.-T., SHEN, D., ZENG, H.-J., YANG, Q., LU, Y., AND CHEN, Z. 2005. Web-page summarization using clickthrough data. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR).* 194–201.

TAO, T., WANG, X., MEI, Q., AND ZHAI, C. 2006. Language model information retrieval with document expansion. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*. 407–414.

TEUFEL, S. AND MOENS, M. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Comput. Ling. 28*, 4, 409–445.

TOMOKIYO, T. AND HURST, M. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. 33–40.

TOUTANOVA, K. AND MANNING, C. D. 2000. Enriching the knowledge sources used in a maximum entropy Part-of-Speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*. 63–70.

TURNEY, P. D. 2000. Learning algorithms for keyphrase extraction. *Inform. Retrieval 2*, 4, 303–336.

TURNEY, P. D. 2003. Coherent keyphrase extraction via web mining. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*. 434–439.

WAN, X. AND XIAO, J. 2008a. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*. 855–860.

WAN, X. AND XIAO, J. 2008b. CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*. 969–976.

WAN, X. AND YANG, J. 2007a. Single document summarization with document expansion. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*. 931–936.

WAN, X. AND YANG, J. 2007b. CollabSum: Exploiting multiple document clustering for collaborative single document summarizations. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 143–150.

WAN, X., YANG, J., AND XIAO, J. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*. 552–559.

WANG, X., SHEN, D., ZENG, H.-J., CHEN, Z., AND MA, W.-Y. 2004. Web page clustering enhanced by summarization. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM)*. 242–243.

WITTEN, I. H., PAYNTER, G. W., FRANK, E., GUTWIN, C., AND NEVILL-MANNING, C. G. 1999. KEA: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries (DL)*. 254–256.

WONG, T.-L., LAM, W., AND CHAN, S.-K. 2006. Collaborative information extraction and mining from multiple web documents. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*. 440–450.

XUE, G.-R., LIN, C., YANG, Q., XI, W., ZENG, H.-J., YU, Y., AND CHEN, Z. 2005. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 114–121.

YIH, W.-T., GOODMAN, J., AND CARVALHO, V. R. 2006. Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web (WWW)*. 213–222.

ZHA, H. Y. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 113–120.

ZHANG, B., LI, H., LIU, Y., JI, L., XI, W., FAN, W., CHEN, Z., AND MA, W.-Y. 2005. Improving web search results using affinity graph. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 504–511.

Received May 2008; revised November 2008, March 2009; accepted April 2009