Incorporating Cross-Document Relationships Between Sentences for Single Document Summarizations

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao

Institute of Computer Science and Technology,
Peking University, Beijing 100871, China
{wanxiaojun, yangjianwu, xiaojianguo}@icst.pku.edu.cn

Abstract. Graph-based ranking algorithms have recently been proposed for single document summarizations and such algorithms evaluate the importance of a sentence by making use of the relationships between sentences in the document in a recursive way. In this paper, we investigate using other related or relevant documents to improve summarization of one single document based on the graph-based ranking algorithm. In addition to the within-document relationships between sentences in the specified document, the cross-document relationships between sentences in different documents are also taken into account in the proposed approach. We evaluate the performance of the proposed approach on DUC 2002 data with the ROUGE metric and results demonstrate that the cross-document relationships between sentences in different but related documents can significantly improve the performance of single document summarization.

1 Introduction

Text summarization is the process of automatically creating a compressed version of a given text that provides useful information for users. Automated text summarization has drawn much attention in recent years because it becomes more and more important in many text applications. For example, current search engines usually provide a short summary for each resultant document so as to facilitate users to browse the results and improve users' search experience. News agents usually provide concise headline news describing hot news and they also produce weekly news review for users, which saves users' time and provide better service quality.

Text summaries can be either query-relevant summaries or generic summaries. A query-relevant summary is usually used in search engines and its content should be closely related to the given query. And a generic summary should contain the main topics of the document while keeping redundancy to a minimum. It is a great challenge to automatically generate a high-quality generic summary for a document without any additional clues and prior knowledge. In this paper, we focus on generic single document summarization.

To the best of our knowledge, almost all previous methods for single document summarization produce a summary for a specified document based only on the information contained in the document. In some cases, a set of related or relevant documents are provided and some single documents in the set are required to be

J. Gonzalo et al. (Eds.): ECDL 2006, LNCS 4172, pp. 403 – 414, 2006.
© Springer-Verlag Berlin Heidelberg 2006

summarized. For example, the documents returned by a search engine for a specified query can be considered topically related to each other. The documents within a cluster produced by a clustering algorithm on a document set are also deemed related and relevant. This study aims to explore whether the cross-document relationships between sentences in different but related documents can contribute to the task of single document summarization. In this paper, we propose the novel idea of incorporating both the cross-document relationships between sentences and the within-document relationships between sentences into the graph-based ranking algorithm for single document summarization. By taking into account these two kinds of relationships between sentences, each sentence in a single document obtains a global ranking score to denote its information richness. Then a greedy algorithm is employed to impose diversity penalty on each sentence of the document based on the overlap between this sentence and other high informative sentences in the document. The sentences with both high information richness and high information novelty are chosen into the single summary for the specified document. We perform experiments on DUC 2002 data and experimental results show that the cross-document relationships between sentences can significantly improve the performance of single document summarization.

The rest of this paper is organized as follows: Section 2 briefly introduces related work. The details of the proposed approach are described in Section 3. Section 4 presents and discusses the evaluation results. Lastly we conclude our paper in Section 5.

2 Related Work

In recent years, single document summarization has been widely explored in the natural language processing and information retrieval communities. A series of workshops and conferences on automatic text summarization (e.g. SUMMAC[1], DUC[2] and NTCIR[3]), special topic sessions in ACL, COLING, and SIGIR have advanced the technology and produced a couple of experimental online systems.

Generally speaking, single document summarization methods can be categorized into two categories: extraction-based methods and abstraction-based methods [9, 10, 13]. Extraction is much easier than abstraction because extraction is just to select existing sentences while abstraction needs sentence compression and reformulation. In this paper, we focus on extraction-based methods.

Extraction-based methods usually assign each sentence a saliency score and then rank the sentences in the document. The scores is usually assigned based on a combination of statistical and linguistic features, including term frequency [17], sentence position [8], cue words [5], stigma words [5], topic signature [16], lexical chains [22], etc. Machine learning methods are also employed to extract sentences, including classification-based methods [1, 14], clustering-based methods [21], HMM-based methods [4], etc. Other methods derived from information retrieval techniques is developed for sentence extraction, including maximal marginal relevance (MMR) [3], latent semantic analysis (LSA) [7], and relevance measure [7].

[1] http://www.itl.nist.gov/iaui/894.02/related_projects/tipster_summac/
[2] http://duc.nist.gov
[3] http://research.nii.ac.jp/ntcir/index-en.html

In [23], the mutual reinforcement principle is employed to iteratively extract key phrases and sentences from a document. Moreover, a method based on text segmentation is proposed by McDonald and Chen [18] and the text segments instead of the sentences are ranked.

Most recently, graph-based ranking methods, including TextRank [19, 20] and LexPageRank [6] have been proposed for document summarization. Similar to PageRank [2] or HITS [12], these methods first build a graph based on the similarity relationships between sentences in a document and then the importance of a sentence is determined by taking into account global information on the graph recursively, rather than relying only on 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 one, it is basically casting a vote for that other sentence. The higher 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.

While in the above graph-based ranking algorithms, each single document is summarized independently, in other words, only sentences within the same document cast votes for each other. We believe that the sentences in other related documents can also cast votes for the sentences in the specified document because for a set of related documents, the information contained in an important sentence of a document will be expressed in other sentences of the other documents. Moreover, if needed, our approach can summarize all single documents in the document set in a batch way.

3 The Proposed Approach

The proposed approach summarizes each single document within a document set based on the graph-based ranking algorithm over all sentences in the document set. The documents in the document set are assumed to be related or relevant[4]. The contribution of the proposed approach is based on the following intuition: The important information expressed in a sentence of a document is also expressed in the sentences of many related documents besides the other sentences within the same document. Figure 1 gives the framework of the proposed approach.

In the framework, the first step aims to build a global affinity graph reflecting the relationships among all sentences in the document set; the second step is to compute information richness of each sentence based on the global affinity graph. The first two steps performs on the whole document set, while the third step performs only on the single document, in other words, the process of information richness computation is on a document set scale and the process of diversity penalty is on a single document scale. We assume that a good summary is expected to include the sentences with both high information richness and high information novelty.

---

[4] As noted in Section 1, we can obtain a set of related documents by clustering algorithms or information retrieval techniques.

1. *Build a global affinity graph G based on all sentences in the document set* $D=\{d_1,d_2,\ldots d_l\}$. *Let the* $S=\{s_1, s_2, \ldots, s_n\}$ *denotes the sentence set.*
2. *Based on the global affinity graph G, the graph-based ranking algorithm is employed to compute a global ranking score* $InfoRich(s_i)$ *for each sentence* s_i, *where* $InfoRich(s_i)$ *denotes the information richness of the sentence* s_i.
3. *for any single document* d_k *to be summarized*

   1) *Extract the local affinity graph* G_{d_k} *for* d_k *from G; Let* S_{d_k} *denotes the set of sentences in* d_k.

   2) *Impose a diversity penalty on each sentence in* S_{d_k} *based on* G_{d_k} *and the obtained global ranking scores of sentences in* S_{d_k}, *and obtain a overall affinity ranking score* $ARScore(s_i)$ *for each sentence* s_i *in* S_{d_k}.

   3) *Choose the sentences with highest overall ranking scores into the summary;*

Fig. 1. The framework of the proposed approach

3.1 Global Affinity Graph Building

Given the sentence collection $S=\{s_i \mid 1 \leqslant i \leqslant n\}$, we measure the similarity between sentences based on co-occurrences of terms in the sentences. Formally, a sentence s_i is represented by the set of N_i words that appear in the sentence: $s_i = w_1^i, w_2^i, \ldots, w_{N_i}^i$. Given two sentences s_i and s_j, the similarity of s_i and s_j is defined as:

$$\mathrm{sim}(s_i, s_j) = \frac{\left|\{w_k \mid w_k \in s_i \ \&\ w_k \in s_j\}\right|}{\left|s_i\right| + \left|s_j\right| - \left|\{w_k \mid w_k \in s_i \ \&\ w_k \in s_j\}\right|} \tag{1}$$

The above measure is known as the Jaccard coefficient [11]. Other sentence similarity measures, such as Cosine similarity, Overlap coefficient, Dice coefficient, etc. are also possible, and we are currently evaluating their impact on the summarization performance.

If sentences are considered as nodes, the sentence collection can be modeled as an undirected graph by generating the edge (link) between two sentences only if their similarity weight exceeds 0, i.e. an undirected link between s_i and s_j ($i \neq j$) with similarity weight $sim(s_i, s_j)$ is constructed if $sim(s_i,s_j)>0$; otherwise no link is constructed.

Thus, we construct an undirected graph G reflecting the relationships between sentences by their content similarity. The graph is called as Affinity Graph. Since the graph contains all sentences in the document set, it is called as Global Affinity Graph.

3.2 Information Richness Computation

The graph-based ranking algorithm [6, 19, 20] is employed to compute information richness of sentences, which is based on the following three intuitions:

1. The more neighbors a sentence has, the more informative it is;
2. The more informative a sentence's neighbors are, the more informative it is.
3. The more heavily a sentence is linked with other informative sentences, the more informative it is.

In previous graph-based ranking algorithms for single document summarization, the neighbors of a sentence all come from the same document, while it is intuitive that the information contained in an informative sentence will be also expressed in the sentences of other related documents and we believe that the votes of neighbors in related documents are also important, so we use both the neighbors from the same document and the neighbors from related documents to iteratively compute the information richness of a sentence.

The graph-based ranking algorithm is similar to PageRank [2]. First, we use an adjacency (affinity) matrix $\mathbf{M}$ to describe the affinity graph with each entry corresponding to the weight of a link in the graph. $\mathbf{M} = (M_{i,j})_{n\times n}$ is defined as follows:

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

In our context, the links (edges) between sentences in the graph can be categorized into two classes: intra-document link and inter-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 called an intra-document link; and if s_i and s_j come from different documents, the link is called an inter-document link. We believe that intra-document links and inter-document links have unequal contributions in the graph based ranking algorithm, so distinct weights are assigned to intra-document links and inter-document links respectively. We decompose the original affinity matrix $\mathbf{M}$ as

$$\mathbf{M} = \mathbf{M}_{\text{intra}} + \mathbf{M}_{\text{inter}} \tag{3}$$

where $\mathbf{M}_{\text{intra}}$ is the affinity matrix containing only the intra-document links (the entries of inter-document links are set to 0) and $\mathbf{M}_{\text{inter}}$ is the affinity matrix containing only the inter-document links (the entries of intra-document links are set to 0).

After we differentiate the intra-document links and inter-document links, the new affinity matrix is as follows:

$$\widehat{\mathbf{M}} = \lambda_1 \mathbf{M}_{\text{intra}} + \lambda_2 \mathbf{M}_{\text{inter}} \tag{4}$$

We let $\lambda_1, \lambda_2 \in [0,1]$ in the experiments. If λ_1 =0 and λ_2 =1, only inter-document links are taken into account in the algorithm, and if λ_1 =1 and λ_2 =0, only intra-document links are taken into account in the algorithm. Note that if $\lambda_1 = \lambda_2$ =1, Equation (4) reduces to Equation (3).

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

$$\tilde{M}_{i,j} = \begin{cases} \widehat{M}_{i,j} \Big/ \sum_{j=1}^{n} \widehat{M}_{i,j}, & \text{if } \sum_{j=1}^{n} \widehat{M}_{i,j} \neq 0 \\ 0 & \text{, otherwise} \end{cases} \tag{5}$$

Note that now we do not have $\tilde{M}_{i,j} = \tilde{M}_{j,i}$ for any pair of *i* and *j*. Based on the normalized adjacency matrix $\tilde{\mathbf{M}} = (\tilde{M}_{i,j})_{n\times n}$, the information richness score for each node can be deduced from those of all other nodes linked with it and it can be formulated in a recursive form as follows:

$$\text{InfoRich}(s_i) = \sum_{\text{all } j\neq i} \text{InfoRich}(s_j) \cdot \tilde{M}_{j,i} \tag{6}$$

The above form can be represented in a matrix form:

$$\vec{\lambda} = \tilde{\mathbf{M}}^T \vec{\lambda} \tag{7}$$

where $\vec{\lambda} = [\text{InfoRich}(s_i)]_{n\times 1}$ is the eigenvector of $\tilde{\mathbf{M}}^T$.

Note that $\tilde{\mathbf{M}}$ is normally a sparse matrix and some rows with all-zero elements could possibly appear because some sentences have no links with other sentences. Similar to the random jumping factor in PageRank, a damping factor *d* (usually 0.85) is introduced in order to compute a reasonable eigenvector:

$$\text{InfoRich}(s_i) = d \cdot \sum_{\text{all } j\neq i} \text{InfoRich}(s_j) \cdot \tilde{M}_{j,i} + \frac{(1-d)}{n} \tag{8}$$

And the matrix form is:

$$\vec{\lambda} = d\tilde{\mathbf{M}}^T \vec{\lambda} + \frac{(1-d)}{n}\vec{e} \tag{9}$$

where $\vec{e}$ is a unit vector with all elements equaling to 1.

The above process can be considered as a Markov chain by taking the sentences as the states and the corresponding transition matrix is given by $d\tilde{\mathbf{M}}^T + (1-d)\mathbf{U}$, where $\mathbf{U} = [\frac{1}{n}]_{n\times n}$. The stationary probability distribution of each state is obtained by the principal eigenvector of the transition matrix.

3.3 Diversity Penalty Imposition

After the information richness of each sentence is computed based on the global affinity graph, we can choose highly informative sentences into the summary for any specified single document in the document set. However, a good summary should keep redundant information as minimal as possible, so we impose a diversity penalty to each sentence. Finally, an overall affinity rank score is obtained to reflect both information richness and information novelty of a sentence in the specified document. Since we aim to produce single document summaries, this diversity penalty process must be applied for each single document separately.

For each single document d_k to be summarized we can extract a sub-graph G_{d_k} only containing the sentences within d_k and the corresponding edges between them from the global affinity graph G. We assume the document d_k has m (m<n) sentences and the sentences' affinity matrix $\mathbf{M}_{\mathrm{d_k}} = (\mathbf{M}_{\mathrm{d_k}})_{\mathrm{m \times m}}$ is derived from the original matrix $\mathbf{M}$ by extracting the corresponding entries. Then $\mathbf{M}_{\mathrm{d_k}}$ is normalized into $\tilde{\mathbf{M}}_{\mathrm{d_k}}$ as Equation (5) to make the sum of each row equal to 1. Similar to [24], a greedy algorithm is used to impose the diversity penalty and compute the final affinity rank score for each sentence within the document. The algorithm goes as follows:

---

1. *Initialize two sets A=ϕ, B={s_i | i=1,2,...,m} for the specified document d_k, and each sentence's overall affinity rank score is initialized to its information richness score, i.e. ARScore(s_i) = InfoRich(s_i), i=1,2,...m.*
2. *Sort the sentences in B by their current affinity rank scores in descending order.*
3. *Suppose s_i is the highest ranked sentence, i.e. the first sentence in the ranked list. Move sentence s_i from B to A, and then a diversity penalty is imposed to the affinity rank score of each sentence linked with s_i in B as follows:*

   For each sentence $s_j \in B$

   $$ARScore(s_j) = ARScore(s_j) - \omega \cdot (\tilde{M}_{\mathrm{d_k}})_{j,i} \cdot InfoRich(s_i)$$

   where ω>0 is the penalty degree factor. The larger ω is, the greater penalty is imposed to the affinity rank score. If ω=0, no diversity penalty is imposed at all.
4. *Go to step 2 and iterate until B=ϕ or the iteration count reaches a predefined maximum number.*

Fig. 2. The algorithm of diversity penalty imposition

In the above algorithm, the third step is the crucial step and its basic idea is to decrease the affinity rank score of less informative sentences by the part conveyed from the most informative one. After the affinity rank scores are obtained for all m sentences in the document d_k, several sentences with highest affinity rank scores are chosen to produce the summary for d_k according to the summary length limit.

The above algorithm is applied once for each single document to be summarized in the document set.

4 Experiments

4.1 Data Set

Single document summarization has been one of the fundamental tasks in DUC 2001 and DUC 2002, i.e. task 1 of DUC 2001 and task 1 of DUC 2002. We used DUC 2001 data for training and DUC 2002 data for testing in the experiments. The task 1 of DUC 2002 aims to evaluate generic summaries with a length of approximately 100 words or less. DUC 2002 provides 567 English news articles for single-document summarization task. The sentences in each article have been separated and the sentence information is stored into files. The 567 articles are collected from TREC-9 and grouped into 59 clusters[5] and the documents within each cluster are related or relevant, so it is appropriate to apply our proposed approach directly. The single summaries for all documents within a cluster are produced in a batch way.

As a preprocessing step, for each document, the dialog sentences (sentences in quotation marks) were removed. The stop words in each sentence were removed and the remaining words were stemmed using the Porter's stemmer[6].

4.2 Evaluation Metric

We use the ROUGE [15] evaluation toolkit[7] for evaluation, which is adopted by DUC for automatic summarization evaluation. It measures 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 is an n-gram recall measure computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{RefSum\}} \sum_{n\text{-}gram \in S} Count_{match}(n-gram)}{\sum_{S \in \{RefSum\}} \sum_{n\text{-}gram \in S} Count(n-gram)} \tag{10}$$

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

ROUGE toolkit reports separate scores for 1, 2, 3 and 4-gram, and also for longest common subsequence co-occurrences. Among these different scores, unigram-based ROUGE score (ROUGE-1) has been shown to agree with human judgment most [15]. We show three of the ROUGE metrics in the experimental results, at a confidence level of 95%: ROUGE-1 (unigram-based), ROUGE-2 (bigram-based), and ROUGE-W (based on weighted longest common subsequence, weight=1.2). Note that we mainly concern ourselves with ROUGE-1 scores.

In order to truncate summaries longer than 100 words, we use the "-l 100" option[8] in ROUGE toolkit and we also use the "-m" option for word stemming.

---

[5] At first, there were 60 document clusters, but the document cluster of D088 is withdrawn by NIST due to differences in documents used by systems and NIST summarizers.

[6] http://www.tartarus.org/martin/PorterStemmer/

[7] We use ROUGEeval-1.4.2 downloaded from http://haydn.isi.edu/ROUGE/

[8] This option is necessary for fair comparison because longer summary will usually increase ROUGE evaluation scores.

4.3 Experimental Results

4.3.1 System Comparison

In the experiments, the proposed system has been compared with top 5 (out of 15) systems and baseline systems. The top five systems are the systems with highest ROUGE scores, chosen from the performing systems on the single document summarization task of DUC 2002. Table 1 shows the system comparison results over three ROUGE metrics[9]. In the table, S21-S31 are the system IDs for the top performing systems. "Intra- & Inter-document link" denotes the proposed approach taking into account both intra-document links between sentences within the specified document and inter-document links between sentences across different but related documents. "Only Inter-document link" and "Only Intra-document link" are two baseline systems: the first one is based only on inter-document links and the second one is based only on intra-document links. Note that previous summarization work [6, 19, 20] using graph-based ranking algorithm is similar to "Only Intra-document link" in this paper. The performance of "Intra- & Inter-document link" is achieved when $\lambda_1=1$ and $\lambda_2=0.7$, $\omega=1$. The performance of "Only Inter- document link" is achieved when $\lambda_1=0$ and $\lambda_2=1$, $\omega=1$. And the performance of "Only Intra-document link" is achieved when $\lambda_1=1$ and $\lambda_2=0$, $\omega=0.5$. Note that the parameters are tuned on DUC 2001 data.

Table 1. System comparison on DUC 2002 data

System	ROUGE-1	ROUGE-2	ROUGE-W
S28	0.48049	0.22832	0.17073
S21	0.47754	0.22273	0.16814
Intra- & Inter-document link	0.47710	0.20457	0.16344
Only Inter-document link	0.47399	0.20332	0.16215
S31	0.46506	0.20392	0.16162
Only Intra-document link	0.46443	0.19072	0.15832
S29	0.46384	0.21246	0.16462
S27	0.46019	0.21273	0.16342

Seen from the table, our proposed system, i.e. "Intra- & Inter-document link", achieves a good performance comparable to that of the state-of-the-art systems, i.e. S28 and S21. The proposed system outperforms both the system based only on the intra-document links (i.e. "Only Intra-document link") and the system based only on the inter-document links (i.e. "Only Inter-document link"), which demonstrates that both the intra-document links and the inter-document links between sentences are important for single document summarization based on the graph-based ranking algorithm. We can also see that the system based only on the inter-document links (i.e. "Only Inter-document link") outperforms the system based only on the intra-

[9] The ROUGE values of top performing systems are different from those reported in [19, 20] because they do not use the "-l 100" option to truncate summaries longer than 100 words.

document links (i.e. "Only Intra-document link"), which further demonstrates the great importance of the cross-document relationships between sentences for single document summarization.

4.3.2 Parameter Tuning

In this section, we investigate tuning the important parameters employed in the proposed systems, including the penalty factor ω for three systems based on graph ranking algorithms, the intra-document link and inter-document link differentiation weights λ_1 and λ_2 for the proposed system, i.e. "Intra- & Inter-document link". The ROUGE-1 results are shown in Figures 3-4 respectively.

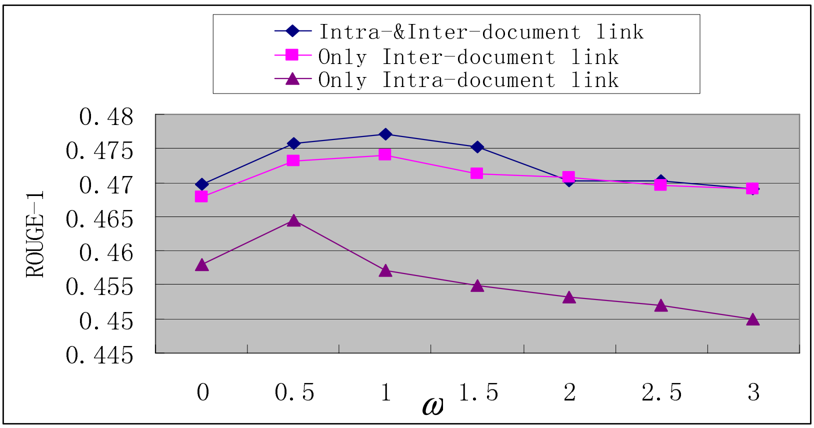

Fig. 3. Penalty factor (ω) tuning for three systems

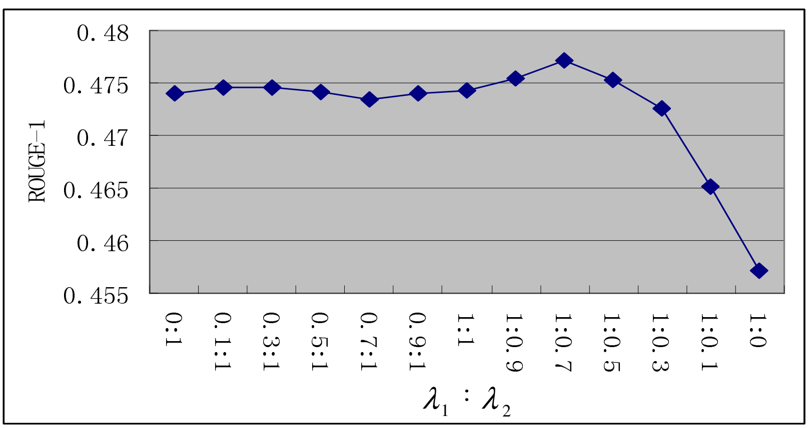

Fig. 4. Intra-document/inter-document link weight ($\lambda_1 : \lambda_2$) tuning for the proposed system (i.e. "Intra- & Inter-document link")

Figure 3 demonstrates the influence of the penalty factor ω for the proposed system when λ_1=1 and λ_2=0.7, and also for the systems of "Only Intra-document link" and "Only Inter-document link". It shows that the proposed system outperforms the two baseline systems over different values of the penalty factor ω. Moreover, the system of "Only Inter-document link" much outperforms the system of "Only Intra-document link" irrespective of the value of ω. We can also see that ω=1 is the point where the proposed system and the system of "Only Inter-document link" achieve their best performances, and more or less diversity penalty will deteriorate their performances.

Figure 4 shows the influence of the intra-document/inter-document link weights λ_1 and λ_2 for the proposed system when ω=1. λ_1 and λ_2 range from 0 to 1. In Figure 4, $\lambda_1 : \lambda_2$ denotes the real values λ_1 and λ_2 are set to. For example, $\lambda_1 : \lambda_2$=1:1 means λ_1=1 and λ_2=1. It is observed that when λ_1=0.3 and λ_2=1 the system can obtain the optimal performance.

5 Conclusion

In this paper, we propose to incorporate cross-document relationships between sentences into the graph-based ranking algorithm for single document summarizations. Experimental results on DUC 2002 data demonstrate the great importance of inter-document links between sentences in different but related documents for single document summarizations based on graph-based ranking algorithm.

References

1. Amini, M. R., Gallinari, P.: The Use of Unlabeled Data to Improve Supervised Learning for Text Summarization. In Proceedings of SIGIR2002, 105-112.
2. Brin, S. and Page, L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 1998, 30:1-7.
3. Carbonell, J., Goldstein, J.: 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, 1998, 335-336.
4. Conroy, J. M., O'Leary, D. P.: Text Summarization via Hidden Markov Models. In Proceedings of SIGIR2001, 406-407.
5. Edmundson, H. P.: New Methods in Automatic Abstracting. Journal of the Association for computing Machinery, 1969, 16(2): 264-285.
6. ErKan, G¨unes, Radev, D. R.: LexPageRank: Prestige in Multi-Document Text Summarization. In Proceedings of EMNLP2004.
7. Gong, Y. H., Liu, X.: Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In Proceedings of SIGIR2001, 19-25.
8. Hovy, E., Lin, C. Y.: Automated Text Summarization in SUMMARIST. In Proceeding of ACL'1997/EACL'1997 Worshop on Intelligent Scalable Text Summarization, 1997.

9. Jing, H.: Sentence Reduction for Automatic Text Summarization. In Proceedings of ANLP 2000.
10. Jing, H., McKeown, K. R.: Cut and Paste Based Text Summarization. In Proceedings of NAACL2000, 178-185.
11. Jones, W. P. and Furnas, G. W. Pictures of relevance: a geometric analysis of similarity measure. Journal of the American Society for Information Science, 1987, 38(6): 420-442.
12. Kleinberg, J. M. Authoritative sources in a hyperlinked environment. Journal of the ACM, 1999, 46(5):604-632.
13. Knight, K., Marcu, D.: Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. Artificial Intelligence, 2002, 139(1): 91-107.
14. Kupiec, J., Pedersen, J., Chen, F.: A.Trainable Document Summarizer. In Proceedings of SIGIR1995, 68-73.
15. Lin, C. Y., Hovy, E.: Automatic Evaluation of Summaries Using N-Gram Co-Occurrence Statistics. In Proceedings of HLT-NAACL2003.
16. Lin, C. Y., Hovy, E.: The Automated Acquisition of Topic Signatures for Text Summarization. In Proceedings of the 17th Conference on Computational Linguistics, 2000, 495-501.
17. Luhn, H. P.: The Automatic Creation of literature Abstracts. IBM Journal of Research and Development, 1969, 2(2).
18. McDonald, D., Chen, H.: Using Sentence-Selection Heuristics to Rank Text Segment in TXTRACTOR. In Proceedings of JCDL2002, 28-35.
19. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In Proceedings of EMNLP2004.
20. Mihalcea, R. and Tarau, P.: A language independent algorithm for single and multiple document summarization. In Proceedings of IJCNLP2005.
21. Nomoto, T., Matsumoto, Y.: A New Approach to Unsupervised Text Summarization. In Proceedings of SIGIR2001, 26-34.
22. Silber, H. G., McCoy, K.: Efficient Text Summarization Using Lexical Chains. In Proceedings of the 5th International Conference on Intelligent User Interfaces, 2000, 252-255.
23. Zha, H. Y.: Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering. In Proceedings of SIGIR2002, 113-120.
24. Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., and Ma, W.-Y. Improving web search results using affinity graph. In Proceedings of SIGIR2005.