CollabSum: Exploiting Multiple Document Clustering for Collaborative 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

Almost all existing methods conduct the summarization tasks for single documents separately without interactions for each document under the assumption that the documents are considered independent of each other. This paper proposes a novel framework called CollabSum for collaborative single document summarizations by making use of mutual influences of multiple documents within a cluster context. In this study, CollabSum is implemented by first employing the clustering algorithm to obtain appropriate document clusters and then exploiting the graph-ranking based algorithm for collaborative document summarizations within each cluster. Both the with-document and cross-document relationships between sentences are incorporated in the algorithm. Experiments on the DUC2001 and DUC2002 datasets demonstrate the encouraging performance of the proposed approach. Different clustering algorithms have been investigated and we find that the summarization performance relies positively on the quality of document cluster.

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, Performance

Keywords: CollabSum, Single document summarization, Collaborative summarization, Graph-ranking algorithm

1. INTRODUCTION

Document summarization is the process of automatically creating a compressed version of a given document that delivers the main topic of the document. Automated document summarization has drawn much attention for a long time 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 portals usually provide concise headline news describing hot news topic each day and they also produce weekly news review to save users' time and improve service quality.

Document summary can be either query-relevant or generic. Query-relevant summary should be closely related to the given query. Generic summary should reflect the main topic of the document without any additional clues and prior knowledge. In this paper, we focus on generic single document summarization.

Very often, all single documents in a document set are required to be summarized. While almost all previous methods for single document summarization produce a summary for a specified document based only on the information contained in the document. One common assumption of existing methods is that the documents are independent of each other. Hence the summarization task is conducted separately without interactions for each document. However, some documents within an appropriate cluster context actually have mutual influence and contain useful clues which can help to extract summary from each other. For example, two documents about the same topic would provide additional knowledge for each other to better evaluate and extract salient information from each other. The idea is borrowed from human's perception that a user would better understand a topic expressed in a document if the user reads another document about the same topic. This study proposes a novel framework called CollabSum for collaborative document summarizations by making use of additional information from multiple documents within appropriate cluster context. The cluster context can be obtained by applying the clustering algorithm on the document set and we have investigated how the cluster context influences the summarization performance by employing different clustering algorithms.

The proposed CollabSum employs the graph-ranking based algorithm for collaborative document summarization of each document in a specified cluster and both the cross-document relationships and the within-document relationships between sentences are incorporated in the algorithm, where the within-document relationships reflect the local information existing in the specified document and the cross-document relationships reflect the global information existing in the cluster context.

We perform experiments on the DUC2001 and DUC2002 datasets and the results demonstrate the good effectiveness of CollabSum. The use of the cross-document relationships between sentences can much improve the performance of single document summarization. We find that the summarization performance is positively correlated with the quality of cluster context and existing clustering algorithms can yield appropriate cluster context for collaborative document summarizations.

The rest of this paper is organized as follows: Section 2 briefly introduces the related work. The proposed CollabSum is described in detail in Section 3. We set up the experiments in Section 4 and give the results in Section 5. Section 6 discusses the results and lastly we conclude this paper in Section 7.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

*This work was supported by the National Science Foundation of China (60642001).

2. RELATED WORK

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. DUC[1] and NTCIR[2]), 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 [11, 12, 14]. 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 score is usually computed based on a combination of statistical and linguistic features, including term frequency [18], sentence position [9], cue words [6], stigma words [6], topic signature [17], lexical chains [25], etc. Machine learning methods are also employed to extract sentences, including classification-based methods [1, 15], clustering-based methods [22], HMM-based methods [5], CRF-based method [24], etc. Other methods include maximal marginal relevance (MMR) [4], latent semantic analysis (LSA) [8], and relevance measure [8]. In [27], 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 [19] and the text segments instead of the sentences are ranked.

Most recently, the graph-ranking based methods, including TextRank [20, 21] and LexPageRank [7], have been proposed for document summarization. Similar to PageRank [3] or HITS [13], these methods first build a graph based on the similarity relationships between 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 a 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 is. 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.

However, all the above methods summarize each single document independently. Particularly, only the sentences within the same document cast votes for each other in the graph-ranking based methods. We believe that the sentences in other topic-related documents can also cast votes for the sentences in the specified document, so both the cross-document relationships and the within-document relationships between sentences are incorporated in the proposed CollabSum in this study.

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

3. THE PROPOSED COLLABSUM

3.1 Overview

Given a document set in which each document needs to be summarized respectively, CollabSum first employs the clustering algorithm (e.g. the agglomerative algorithm, the divisive algorithm, the k-means algorithm, etc.) [10, 26] to group the documents into a few clusters. The documents within each cluster are expected to be topic-related and each cluster can be considered as a context for any document in the cluster. Given a document cluster, CollabSum incorporates both the within-document relationships (local information) and the cross-document relationships (global information) between sentences into the graph-ranking based algorithm to summarize each single document within the cluster. Figure 1 gives the framework of the proposed approach.

1. **Document Clustering**: *Group the documents in the document set into a few clusters using the clustering algorithm;*
2. **Document Summarization**: *For each cluster, perform the following steps respectively to produce summaries for single documents in the cluster:*

   1) **Affinity Graph Building***: Build a global affinity graph G based on all sentences in the documents of the given cluster: $D=\{d_1,d_2,...d_l\}$, where l is the number of documents. Let $S=\{s_1, s_2, ..., s_n\}$ denote the sentence set for the cluster, where n is the number of sentences.*

   2) **Informativeness Score Computation***: Based on the global affinity graph G, the graph-ranking based 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) **Within-Document Redundancy Removing***: For any single document d_k to be summarized, the greedy algorithm is employed to remove redundancy for the informative sentences. Finally, the sentences which are both informative and novel are chosen into the summary.*

Figure 1: The framework of CollabSum

For the first step of the above framework, different clustering algorithms will yield different clusters and the documents in a high-quality cluster are usually deemed to be highly topic-related (i.e. appropriate cluster context), while the documents in a low-quality cluster are usually not topic-related (i.e. inappropriate cluster context). The quality of a cluster will influence the reliability of the contextual information for evaluating the importance of the sentences in the cluster. A number of clustering algorithms will be investigated in the experiments.

For the second step of the above framework, step 1) aims to build a global affinity graph reflecting the relationships among all sentences in the document set of the given cluster. Step 2) aims to compute the informativeness score of each sentence based on the global affinity graph. The informativeness of a sentence indicates how much information about the main topic the sentence contains. Step 3) aims to remove redundant information in the summary and keep the sentences in the summary as novel as possible. Step 1) and 2) perform on all documents in the cluster in order to get highly informative sentences from a global perspective, while step

3) performs only on each single document in order to remove redundancy from a local perspective. A summary is expected to include the sentences which are both highly informative and highly novel. Note that the summarization tasks are conducted in a batch mode for each cluster. The steps of 1), 2) and 3) will be described in next sections respectively.

3.2 Affinity Graph Building

Given a sentence collection $S=\{s_i \mid 1\le i\le n\}$ of a specified cluster, the affinity weight $sim(s_i, s_j)$ between a sentence pair of s_i and s_j is calculated using the Cosine measure [2]. The weight associated with term t is calculated with the $tf_t.isf_t$ formula, where tf_t is the frequency of term t in the sentence and isf_t is the inverse sentence frequency of term t, i.e. $1+log(N/n_t)$, where N is the total number of sentences in a background corpus and n_t is the number of sentences containing term t.

If sentences are considered as nodes, the sentence collection can be modeled as an undirected graph by generating a link between two sentences if their affinity weight exceeds 0, i.e. an undirected link between s_i and s_j ($i\neq j$) with the affinity 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 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 link reflects the local information in a document, while the cross-document link reflects the global information in a cluster context, which is exploited by CollabSum to make use of mutual influences between different documents in the cluster. The graph G contains both kinds of links between sentences and is called as *Global Affinity Graph*. We use an adjacency (affinity) matrix $\mathbf{M}$ to describe G 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:

$$M_{i,j} = \begin{cases} sim(s_i,s_j), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

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

$$\tilde{M}_{i,j} = \begin{cases} M_{i,j} \Big/ \sum_{j=1}^{n} M_{i,j}, & \text{if } \sum_{j=1}^{n} M_{i,j} \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Similar to the above process, another two affinity graphs G_{intra} and G_{inter} are also built: the within-document affinity graph G_{intra} is to include 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} is to include 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 $\mathbf{M}_{\text{intra}}$ and $\mathbf{M}_{\text{inter}}$ respectively. $\mathbf{M}_{\text{intra}}$ and $\mathbf{M}_{\text{inter}}$ can be extracted from $\mathbf{M}$ and we have $\mathbf{M}=\mathbf{M}_{\text{intra}}+\mathbf{M}_{\text{inter}}$. Similar to Equation (2), $\mathbf{M}_{\text{intra}}$ and $\mathbf{M}_{\text{inter}}$ are respectively normalized to $\tilde{\mathbf{M}}_{\text{intra}}$ and $\tilde{\mathbf{M}}_{\text{inter}}$ to make the sum of each row equal to 1.

3.3 Informativeness Score Computation

Based on the global affinity graph G, the informativeness score $IFScore_{all}(s_i)$ for sentence s_i can be deduced from those of all other sentences linked with it and it can be formulated in a recursive form as follows[7, 20, 21, 28]:

$$IFScore_{all}(s_i) = d \times \sum_{all\ j\neq i} IFScore_{all}(s_j) \times \tilde{M}_{j,i} + \frac{(1-d)}{n} \quad (3)$$

And the matrix form is:

$$\vec{\lambda} = d\tilde{\mathbf{M}}^T \vec{\lambda} + \frac{(1-d)}{n}\vec{e} \quad (4)$$

where $\vec{\lambda} = [IFScore_{all}(s_i)]_{n\times 1}$ is the vector of informativeness scores. $\vec{e}$ is a unit vector with all elements equaling to 1. d is the damping factor usually set to 0.85.

For implementation, the initial informativeness scores of all sentences are set to 1 and the iteration algorithm in Equation (3) 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:

$$IFScore_{intra}(s_i) = d \times \sum_{all\ j\neq i} IFScore_{intra}(s_j) \times (\tilde{M}_{intra})_{j,i} + \frac{(1-d)}{n} \quad (5)$$

$$IFScore_{inter}(s_i) = d \times \sum_{all\ j\neq i} IFScore_{inter}(s_j) \times (\tilde{M}_{inter})_{j,i} + \frac{(1-d)}{n} \quad (6)$$

The final informativeness score $IFScore(s_i)$ of sentence s_i can be either $IFScore_{all}(s_i)$, $IFScore_{intra}(s_i)$ or $IFScore_{inter}(s_i)$, or the linear combination of $IFScore_{intra}(s_i)$ and $IFScore_{inter}(s_i)$ as follows:

$$IFScore(s_i) = \lambda \cdot IFScore_{intra}(s_i) + (1-\lambda) \cdot IFScore_{inter}(s_i) \quad (7)$$

where $\lambda \in [0,1]$ is a weighting parameter, specifying the relative contributions to the final informativeness scores from the cross-document relationships and the within-document relationships between sentences. If $\lambda=0$, $IFScore(s_i)$ is equal to $IFScore_{inter}(s_i)$; if $\lambda=1$, $IFScore(s_i)$ is equal to $IFScore_{intra}(s_i)$; and if $\lambda=0.5$, the cross-document relationships and the within-document relationships are assumed to be equally important.

We will investigate all the above methods for informativeness score computation. Note that all previous graph-ranking based methods do not consider the cross-document links and have $IFScore(s_i)= IFScore_{intra}(s_i)$.

3.4 Within-Document Redundancy Removing

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 $\mathbf{M}_{d_k} = (M_{d_k})_{m\times m}$ is derived from the original matrix $\mathbf{M}$ by extracting the corresponding entries. Then

$\mathbf{M}_{d_k}$ is normalized into $\widetilde{\mathbf{M}}_{d_k}$ as Equation (2) to make the sum of each row equal to 1. Similar to [28], the greedy algorithm is used to penalize the sentences highly overlapping with other informative sentences based on $\widetilde{\mathbf{M}}_{d_k}$. Finally, the overall rank score for each sentence within the document is obtained and the sentences with highest overall rank scores are both highly informative and highly novel, which are chosen into the summary for d_k according to the summary length limit.

The basic idea of the algorithm is to decrease the overall rank score of less informative sentences by the part conveyed from the most informative one. The overall rank score $ORScore(s_i)$ of any sentence s_i is initialized to its informativeness score. Once the highest ranked sentence s_i is chosen into the summary, any remaining sentence s_j linked with s_i are penalized as follows:

$$ORScore(s_j) = ORScore(s_j) - (\widetilde{M}_{d_k})_{j,i} \times IFScore(s_i) \quad (8)$$

The details of the algorithm are omitted due to page limit. The algorithm is applied once for each single document to be summarized in the document cluster.

4. EXPERIMENTAL SETUP

4.1 Data Set

We use the DUC2001 and DUC2002 datasets for evaluation in the experiments. Both task 1 of DUC2001 and task 1 of DUC 2002 aim to evaluate generic single document summaries with a length of approximately 100 words or less. Table 1 gives a short summary of the two datasets. The sentences in each article have been separated and the sentence information is stored into files. The articles have been grouped into clusters manually and the documents within each cluster are topic-related or relevant. The manually labeled clusters are considered as the ground truth clusters or gold clusters. In order to investigate different clustering algorithms, the documents in the clusters are mixed together to form the whole document set for single document summarizations. As a preprocessing step, the stop words in each sentence are removed and the remaining words are stemmed using the Porter's stemmer [23].

Table 1: 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

4.2 Document Clustering Algorithms

In the experiments, several popular clustering algorithms and random clustering algorithms are explored to produce cluster contexts. Note that we know the numbers of the clusters for the two datasets beforehand and simply use them as input for the following clustering algorithms[3].

Gold Clustering: It is a pseudo clustering algorithm by manually grouping the documents. For any of the two datasets, we use the ground truth clusters as the upperbound of the automatic clustering algorithms.

Agglomerative (AverageLink) Clustering: It is a bottom-up hierarchical clustering algorithm and starts with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters, until the number of the clusters reduces to the desired number. The similarity between two clusters is computed using the AverageLink method, which computes the average of the Cosine similarity values between any pair of documents belonging to the two clusters respectively as follows:

$$sim(c_1,c_2) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} sim(d_i,d_j)}{m \times n} \quad (9)$$

where d_i, d_j are two documents in cluster c_1 and cluster c_2 respectively, and m is the number of documents in cluster c_1 and n is the number of document in cluster c_2.

Agglomerative (CompleteLink) Clustering: It differs from the above agglomerative (AverageLink) clustering algorithm only in that the similarity between two clusters is computed using the CompleteLink method, which computes the minimum of the Cosine similarity values between any pair of documents belonging to the two clusters respectively as follows:

$$sim(c_1,c_2) = \min_{d_i \in c_1, d_j \in c_2} \{sim(d_i,d_j)\} \quad (10)$$

Divisive Clustering: It is a top-down hierarchical clustering algorithm and starts with one, all-inclusive cluster and, at each step, splits the largest cluster (i.e. the cluster with most documents) into two small clusters using the KMeans algorithm until the number of clusters increases to the desired number.

KMeans Clustering: It is a partition based clustering algorithm. The algorithm randomly selects k documents as the initial centroids of the k clusters and then iteratively assigns all documents to the closest cluster, and recomputes the centroid of each cluster, until the centroids do not change. The similarity between a document and a cluster centroid is computed using the standard Cosine measure.

Random1 Clustering: It produces k clusters by randomly assigning each document into one of the k clusters.

Random2 Clustering: It randomly produces k clusters in a different randomization process.

Random3 Clustering: It randomly produces k clusters in another different randomization process.

4.3 Implemented Summarization Systems

Given a cluster of documents, we can design the following three summarization methods based on how to use the cross-document relationships between sentences in the cluster for computing the informativeness scores of sentences:

UniformLink: The method computes the informativeness score of a sentence based on the global affinity graph with both the cross-document relationships and the within-document relationships between sentences, i.e. $IFScore(s_i) = IFScore_{all}(s_i)$;

InterLink: The method computes the informativeness score of a sentence based only on the cross-document relationships between sentences, i.e. $IFScore(s_i) = IFScore_{inter}(s_i)$;

UnionLink: The method computes the informativeness scores $IFScore_{inter}(s_i)$ and $IFScore_{intra}(s_i)$ of sentence s_i based on the cross-document relationships and the within-document

[3] How to obtain the number of desired clusters is not the focus of this study.

relationships between sentences respectively, and then combines them as in Equation (7) to get the final informativeness score. Typically, we let λ=0.5 to make the two kind of relationships equally important. i.e., $IFScore(s_i) = 0.5 * IFScore_{intra}(s_i) + 0.5 * IFScore_{inter}(s_i)$;

In addition, we design the following baseline summarization method using only the within-document relationships between sentences in a document, which is widely explored in previous work [7, 20, 21].

IntraLink: The method computes the informativeness score of a sentence based only on the within-document relationships between sentences, i.e. $IFScore(s_i) = IFScore_{intra}(s_i)$;

The cross-document methods of "InterLink", "UnionLink" and "UniformLink" rely on the clustering algorithm adopted for document clustering, and a summarization system implementing CollabSum is represented by a combination of one of the clustering algorithms and one of the above cross-document summarization methods. The system based on the "IntraLink" method is the baseline summarization system, which is independent of any clustering algorithm. Note that the process of redundancy removing is the same for all the above methods.

4.4 Evaluation Metric

4.4.1 Document Clustering Evaluation

We adopt the widely used F-Measure to evaluate the performance of the clustering algorithm (i.e. the quality of the clusters) by comparing the produced clusters with the gold clusters (classes) as follows [10]:

For cluster j and class i, we have $Recall(i,j)=n_{ij}/n_i$, $Precision(i,j)=n_{ij}/n_j$, where n_{ij} is the number of members of class i in cluster j, n_j is the number of members of cluster j and n_i is the number of members of class i. The F-Measure of cluster j and class i is then given by

$$F(i,j) = (2*Precision(i,j)*Recall(i,j))/(Precision(i,j)+Recall(i,j))$$

For an entire clustering, the F-measure of any class is the maximum value it attains at any cluster and an overall value for the F-measure is computed by taking the weighted average of all values for the F measure as

$$F = \sum_i \frac{n_i}{n} \max_j \{F(i,j)\} \quad (11)$$

where the max is taken over all clusters and n is the number of all documents in the set. The larger the F-Measure is, the better the cluster quality is.

4.4.2 Document Summarization Evaluation

We use the ROUGE [16] toolkit (i.e. ROUGEeval-1.4.2 in this study) for evaluation, which is widely 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:

$$ROUGE\text{-}N = \frac{\sum_{S \in \{RefSum\}} \sum_{n\text{-}gram \in S} Count_{match}(n\text{-}gram)}{\sum_{S \in \{RefSum\}} \sum_{n\text{-}gram \in S} Count(n\text{-}gram)} \quad (12)$$

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, the unigram-based ROUGE score (ROUGE-1) has been shown to agree with human judgment most [16]. We show 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).

In order to truncate summaries longer than length limit, we use the "-l" option[4] in ROUGE toolkit.

5. EXPERIMENTAL RESULTS

First of all, we show the document clustering results in Table 2. The gold clustering result is the upperbound of all automatic clustering results. Seen from the table, the four popular clustering algorithms (i.e. CompleteLink, AverageLink, KMeans and Divisive) all perform much better than the three random clustering algorithms (i.e. Random1, Random2 and Random3). Different clustering results lead to different document relationships and a high-quality cluster produced by popular algorithms is deemed to build an appropriate cluster context for collaborative summarizations.

Now we compare the summarization results on the two datasets in Tables 3 and 4 respectively. In the tables, "IntraLink" is the baseline system and all the other systems are specific implementations of CollabSum. For example, "InterLink (Gold)" is implemented by using the gold clustering for document clustering and the "InterLink" method for summarization. The systems in the tables are listed in a decreasing order of the ROUGE-1 value.

Seen from the tables, most proposed systems using the popular clustering algorithm or gold clustering algorithm outperform the baseline "IntraLink". The systems of "UniformLink (Gold)" and "UnionLink (Gold)", which make use of both the within-document relationships and the cross-document relationships betweens sentences in the ideal (gold) clusters, almost perform best on both datasets, except for "UniformLink(Gold)" on the DUC2001 dataset. It is encouraging that the "UniformLink" and "UnionLink" methods using the popular clustering algorithms always outperform the baseline "IntraLink", which demonstrates that the mutual influences through the cross-document relationships between sentences within a high-quality cluster do benefit single document summarizations. The importance of the cross-document relationships are further validated by that even the methods considering only the cross-document relationships between sentences ("InterLink"), based on high-quality clusters, can perform better than or at least comparable to the baseline "IntraLink".

We can also observe that all the proposed systems using the random clustering algorithms (i.e. the random1, random2 and random3 algorithms) perform not well, even much worse than the

[4] The "-l" option is very important for fair comparison. Some previous works do not adopt this option, which is likely to overestimate the ROUGE scores.

baseline "IntraLink" system in most cases. This is because that the random clustering algorithms usually produce low-quality clusters, in which the documents are not truly topic-related, and so in any of these clusters, the mutual influences through the cross-document relationships between sentences are not reliable for evaluating sentences.

Table 2: Clustering results

Clustering Algorithm	F-Measure	
	DUC2001	DUC2002
Gold	1.000	1.000
CompleteLink	0.907	0.799
AverageLink	0.877	0.752
Divisive	0.924	0.752
K-Means	0.866	0.722
Random1	0.187	0.168
Random2	0.189	0.168
Random3	0.183	0.167

Table 3: System comparison on DUC 2001

System	ROUGE-1	ROUGE-2	ROUGE-W
UnionLink (Gold)	0.44038*	0.16229*	0.13678
UnionLink (AverageLink)	0.43950*	0.16108	0.13679
UnionLink (CompleteLink)	0.43947*	0.16172*	0.13701*
UnionLink (KMeans)	0.43895	0.16054	0.13623
UniformLink (Gold)	0.43890*	0.16213*	0.13676
UnionLink (Divisive)	0.43832	0.15988	0.13598
UniformLink (CompleteLink)	0.43777	0.16097	0.13646
UniformLink (KMeans)	0.43726	0.15990	0.13612
UniformLink (AverageLink)	0.43651	0.15989	0.13592
InterLink (CompleteLink)	0.43556	0.15993	0.13547
UniformLink (Divisive)	0.43524	0.15846	0.13522
InterLink (Gold)	0.43422	0.15872	0.13506
IntraLink	0.43407	0.15696	0.13629
InterLink (KMeans)	0.43402	0.15769	0.13511
InterLink (AverageLink)	0.43397	0.15875	0.13503
InterLink (Divisive)	0.43371	0.15787	0.13461
UniformLink(Random3)	0.43301	0.15678	0.13446
UniformLink(Random1)	0.43269	0.15501	0.13381
UniformLink(Random2)	0.43191	0.15340	0.13393
UnionLink (Random2)	0.43182	0.15228	0.13360
UnionLink (Random3)	0.43099	0.15294	0.13364
UnionLink (Random1)	0.43006	0.15291	0.13316
InterLink (Random1)	0.41016	0.13660	0.12643
InterLink (Random2)	0.40835	0.13481	0.12563
InterLink (Random3)	0.40944	0.13728	0.12588

Table 4: System comparison on DUC 2002

System	ROUGE-1	ROUGE-2	ROUGE-W
UniformLink (Gold)	0.47187*	0.20102*	0.16318
UnionLink (Gold)	0.47028*	0.20046*	0.1626
UnionLink (CompleteLink)	0.46981*	0.19985*	0.16253
UnionLink (AverageLink)	0.46936*	0.19957*	0.16241
UniformLink (CompleteLink)	0.46902*	0.19833	0.16192
UnionLink (Divisive)	0.46844	0.19870*	0.16168
UniformLink (AverageLink)	0.46825*	0.19724	0.16211
InterLink (Gold)	0.46776	0.19628	0.16122
UniformLink (Divisive)	0.46754	0.19696	0.16165
InterLink (KMeans)	0.46749	0.19616	0.16133
InterLink (CompleteLink)	0.46665	0.19639	0.16153
UniformLink (KMeans)	0.46618	0.19603	0.16062
UnionLink (KMeans)	0.46603	0.19630	0.16102
InterLink (AverageLink)	0.46577	0.19457	0.16091
InterLink (Divisive)	0.46504	0.19411	0.16045
IntraLink	0.46261	0.19457	0.16018
UniformLink(Random2)	0.46102	0.19068	0.15849
UniformLink(Random3)	0.46019	0.18904	0.15742
UniformLink(Random1)	0.45942	0.18766	0.15801
UnionLink (Random3)	0.45876	0.18821	0.15761
UnionLink (Random1)	0.45852	0.18676	0.15785
UnionLink (Random2)	0.45812	0.18645	0.15740
InterLink (Random2)	0.43468	0.16706	0.14716
InterLink (Random3)	0.43378	0.16793	0.14782
InterLink (Random1)	0.43138	0.16275	0.14654

(* indicates that the improvement over "IntraLink" is statistically significant.)

In order to better understand the results, the performances of the proposed systems are visually compared in Figures 2-3. We only show ROUGE-1 results due to page limit. The clustering algorithms in *x*-axis are arranged in an ascending order of their F-measure values. We can clearly see that high-quality clusters lead to high summarization performances. The proposed systems using the popular clustering algorithms perform much better than the systems using the random clustering algorithms.

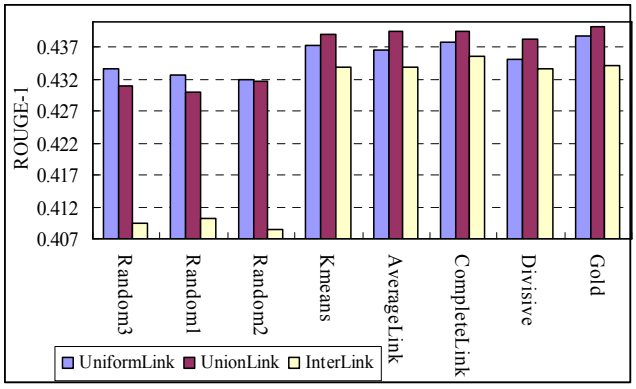

Figure 2: ROUGE-1 vs. clustering algorithm on DUC2001

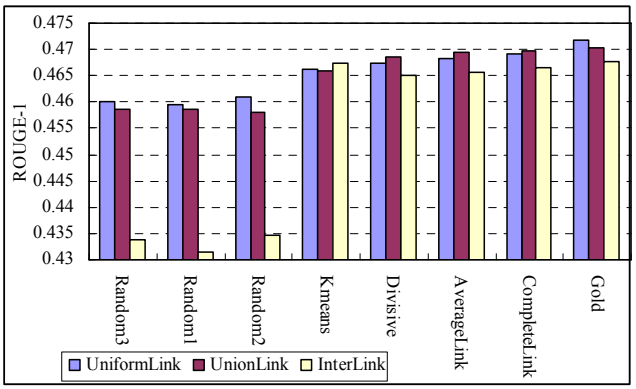

Figure 3: ROUGE-1 vs. clustering algorithm on DUC2002

In order to investigate how the relative contributions from the cross-document relationships and the within-document relationships between sentences influence the summarization performance, Figures 4-7 show the ROUGE-1 values of the systems based on the "UnionLink" method with respect to different values of the combing weight λ. Figures 4 and 5 show the performances of the systems using the popular clustering algorithms and Figures 6 and 7 shows the performances of the systems using the random clustering algorithms. With the increase of λ, the cross-document relationships between sentences contribute less to the final summarization performance, and the within-document relationships between sentences contribute more to the final summarization performance. Seen from Figures 4 and 5, when using the popular clustering algorithms for document clustering, the ROUGE values of the systems first increase and then decrease with the increase of λ, and the best performances are achieved by assigning appropriate relative contributions of the cross-document and within-document relationships between sentences. Seen from Figures 6 and 7, when using the random clustering algorithms for document clustering, the ROUGE values of the systems almost always increase with λ, i.e., the reduction of the contribution of the cross-document relationships can improve the summarization performance, because the random clustering algorithms usually produce low-quality clusters, in which the documents are not topic-related, so the cross-document relationships between sentences in the clusters could not be considered as reliable evidences for evaluating the importance of the sentences.

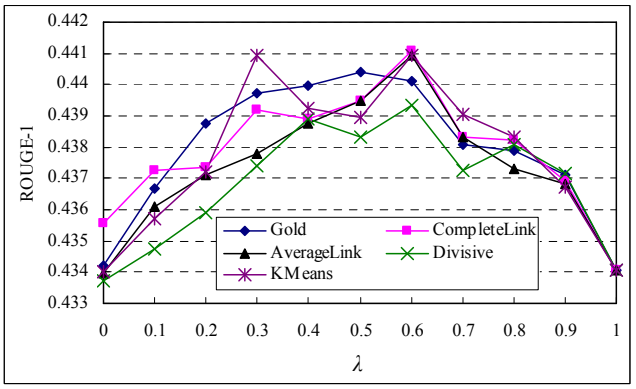

Figure 4: ROUGE-1 vs. λ on high-quality clusters of DUC2001

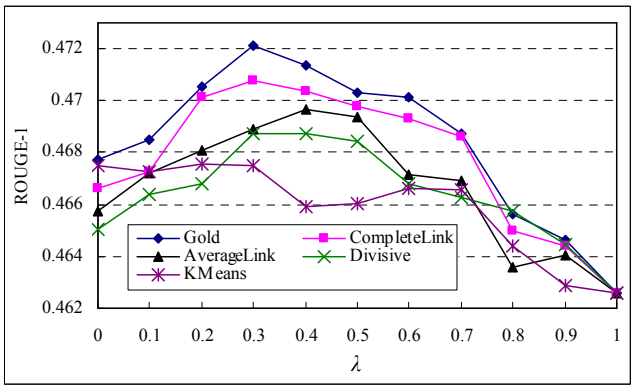

Figure 5: ROUGE-1 vs. λ on high-quality clusters of DUC2002

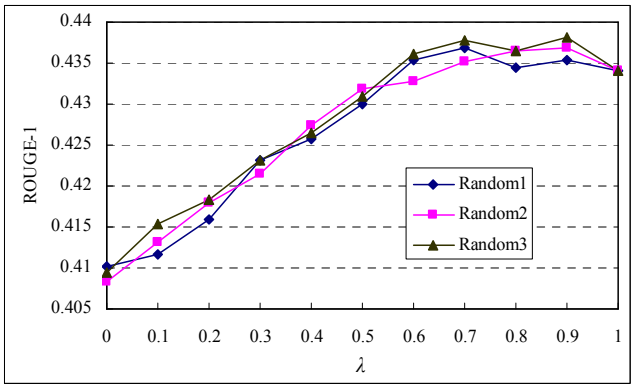

Figure 6: ROUGE-1 vs. λ on low-quality clusters of DUC2001

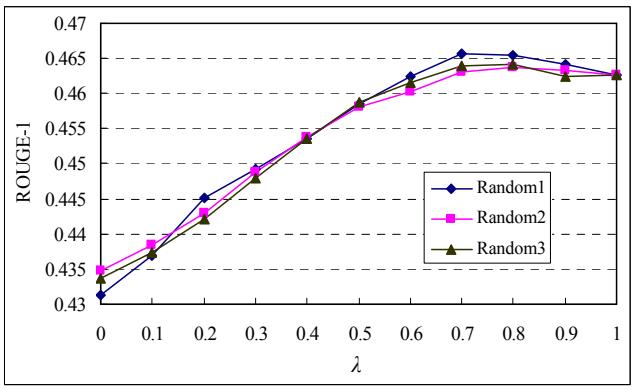

Figure 7: ROUGE-1 vs. λ on low-quality clusters of DUC2002

6. DISCUSSION

The reason underlying the above observations that the cross-document relationships in the framework of CollabSum can improve single document summarizations is that the adopted graph-ranking based algorithm evaluates the importance of a sentence based on the "recommendation" and "voting" from its neighboring sentences. We believe that the votes of neighbors in an appropriate cluster context are at least as important as the votes of neighbors in the same document, so we use both the neighbors from the same document and the neighbors from other documents to iteratively compute the informativeness score of a sentence. In the real world, information usually redundantly exists, for example,

there are many different documents on the Internet to discuss the same topic from various perspectives, and users can obtains thousands of documents for a specified topic through search engines. An important piece of information about a topic in a sentence would be expressed in different ways in the sentences of other documents, and the sentences might have different representations. The appropriate cluster context would guarantee that the mutual influences through the cross-document relationships between sentences are reliable. The proposed approach thus makes use of this phenomenon to incorporate the guaranteed cross-document relationships between sentences for collaborative single document summarizations.

7. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel framework-CollabSum for collaborative single document summarizations, which first groups the documents into clusters and then incorporates the cross-document relationships between sentences in a cluster into the graph-ranking based summarization algorithm. Experimental results on the DUC2001 and DUC2002 datasets demonstrate the good effectiveness of CollabSum. The cross-document relationships between sentences in an appropriate cluster context can improve single document summarizations. The clustering algorithm is important for obtaining the appropriate cluster context and the low-quality clustering results will deteriorate the summarization performance. It is encouraging that most existing popular clustering algorithms can meet the demands of the proposed approach.

The proposed CollabSum has more implementations than the graph-ranking based implementations in this study. In future work, we will explore more summarization methods in the proposed framework to validate the robustness of the framework. Furthermore, we will adapt the proposed approach to collaboratively summarize single web pages. Web pages have rich link information and we can make use of the link structure between web pages to find the appropriate cluster context and mine the implicit mutual influences between web page units for single page summarizations.

8. 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] Baeza-Yates, R., and Ribeiro-Neto, B. Modern Information Retrival. ACM Press and Addison Wesley, 1999.

[3] Brin, S. and Page, L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 1998, 30:1-7.

[4] 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.

[5] Conroy, J. M., O'Leary, D. P.: Text Summarization via Hidden Markov Models. In Proceedings of SIGIR2001, 406-407.

[6] Edmundson, H. P.: New Methods in Automatic Abstracting. Journal of the Association for computing Machinery, 1969, 16(2): 264-285.

[7] ErKan, G¨unes, Radev, D. R.: LexPageRank: Prestige in Multi-Document Text Summarization. In Proceedings of EMNLP2004.

[8] Gong, Y. H., Liu, X.: Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In Proceedings of SIGIR2001, 19-25.

[9] Hovy, E., Lin, C. Y.: Automated Text Summarization in SUMMARIST. In Proceeding of ACL'1997/EACL'1997 Worshop on Intelligent Scalable Text Summarization, 1997.

[10] Jain, A. K, Murty, M. N., and Flynn, P. J. Data clustering: a review. ACM Computing Surveys, 31(3):264-323, 1999.

[11] Jing, H.: Sentence Reduction for Automatic Text Summarization. In Proceedings of ANLP 2000.

[12] Jing, H., McKeown, K. R.: Cut and Paste Based Text Summarization. In Proceedings of NAACL2000, 178-185.

[13] Kleinberg, J. M. Authoritative sources in a hyperlinked environment. Journal of the ACM, 1999, 46(5):604-632.

[14] Knight, K., Marcu, D.: Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. Artificial Intelligence, 2002, 139(1): 91-107.

[15] Kupiec, J., Pedersen, J., Chen, F.: A.Trainable Document Summarizer. In Proceedings of SIGIR1995, 68-73.

[16] Lin, C. Y., Hovy, E.: Automatic Evaluation of Summaries Using N-Gram Co-Occurrence Statistics. In Proceedings of HLT-NAACL2003.

[17] 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.

[18] Luhn, H. P.: The Automatic Creation of literature Abstracts. IBM Journal of Research and Development, 1969, 2(2).

[19] McDonald, D., Chen, H.: Using Sentence-Selection Heuristics to Rank Text Segment in TXTRACTOR. In Proceedings of JCDL2002, 28-35.

[20] Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In Proceedings of EMNLP2004.

[21] Mihalcea, R. and Tarau, P.: A language independent algorithm for single and multiple document summarization. In Proceedings of IJCNLP2005.

[22] Nomoto, T., Matsumoto, Y.: A New Approach to Unsupervised Text Summarization. In Proceedings of SIGIR2001, 26-34.

[23] Porter, M. F. An algorithm for suffix stripping. Program, 14(3): 130-137, 1980.

[24] Shen, D., Sun, J.-T., Li, H., Yang, Q., and Chen, Z. Document Summarization using Conditional Random Fields. In Proceedings of IJCAI 2007.

[25] 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.

[26] SteinBack, M., Karypis, G., and Kumar, V. A comparison of document clustering techniques. In KDD Workshop on Text Mining, 1999.

[27] Zha, H. Y.: Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering. In Proceedings of SIGIR2002, 113-120.

[28] 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.