Update Summarization via Graph-Based Sentence Ranking

Xuan Li, Liang Du, and Yi-Dong Shen

Abstract—Due to the fast evolution of the information on the Internet, update summarization has received much attention in recent years. It is to summarize an evolutionary document collection at current time supposing the users have read some related previous documents. In this paper, we propose a graph-ranking-based method. It performs constrained reinforcements on a sentence graph, which unifies previous and current documents, to determine the salience of the sentences. The constraints ensure that the most salient sentences in current documents are updates to previous documents. Since this method is NP-hard, we then propose its approximate method, which is polynomial time solvable. Experiments on the TAC 2008 and 2009 benchmark data sets show the effectiveness and efficiency of our method.

Index Terms—Summarization, update summarization, topic-focused summarization, multidocument summarization, extraction-based summarization, graph-based ranking, manifold ranking, large-margin constrained ranking, novelty, quadratically constrained quadratic programming, quadratic programming

---

1 INTRODUCTION

UPDATE summarization[1] was proposed by the Document Understanding Conferences[2] (DUC) in 2007. It can be described as follows: given an evolutionary document collection and a topic, generate a topic-relevant summary for the document collection at current time assuming the users have read the earlier documents. A summary consists of a set of sentences in the documents. A good summary is supposed to have the following four properties [4], [9]:

1. **Salience**. It should cover the main contents of the documents.
2. **Relevance**. It should be relevant to the topic so that it can meet the information need of the users.
3. **Nonredundancy**. It contains no redundant contents.
4. **Update**. It should be an update of the previous documents, excluding contents that have already been read by the users.

In this paper, by document summarization, we refer to nonupdate summarization. The first three properties above are also requirements for topic-focused document summarization [30], [32], [35]. Only the last one is a new requirement specific to update summarization.

Graph-ranking-based methods are widely used for update summarization. PNR[2] (Li et al. [17]) and Manifold Ranking with Sink Point (MRSP) (Du et al. [9]) are two representative ones. They build a unified sentence graph from current and previous documents, in which reinforcements between sentences are used to determine their scores and the highest scored sentences are extracted to form a summary. PNR[2] applies negative reinforcements to PageRank, while MRSP uses reinforcement losses for Manifold Ranking. Previous documents directly participate in the reinforcement propagation of the current documents. Since reinforcement propagation is to determine sentence salience, one consequent problem is that salience of the sentences in current documents may be disturbed by previous documents.

To handle the above problem, in this paper, we propose a new graph-ranking based method, called QCQPSum, for update summarization. The key idea of this method is that it treats property 4 as inequality constraints and performs a constrained reinforcement process to determine sentence salience. In the reinforcement process, previous documents act as constraints without directly participate in the reinforcement propagation in current documents. Therefore, the problem mentioned above would not occur. Specifically, QCQPSum is an optimization problem w.r.t. the scores of the sentences. The objective function of QCQPSum mainly consists of four terms: two regularization terms for previous and current documents, respectively, to determine sentence salience, a topic fitting term to keep topic relevant and a suppression term for the update property. Also a set of inequality constraints for the update property are added. Combining these terms leads to the optimization problem of QCQPSum.

The optimization problem is a quadratically constrained quadratic programming (QCQP) problem. Since QCQP problems are NP-hard, we propose an approximate method (named QPSum) which formulates the problem as a convex quadratic programming (QP) problem. QPSum is solvable in polynomial time. Experiments on the TAC[3] 2008 and

---

1. http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html.
2. http://duc.nist.gov/.

- *The authors are with the State Key Lab of Computer Sciences, Institute of Software, Chinese Academy of Sciences, #4 South Fourth Street, Zhongguancun, Haidian District, Beijing 100190, China. E-mail: {lixuan, duliang, ydshen}@ios.ac.cn.*

3. DUC became a Summarization track in the Text Analysis Conference (TAC) from 2008.

2009 benchmark data sets show that it significantly outperforms PNR^2 and MRSP, and achieves comparable efficiency to the state-of-the-art graph-ranking-based methods.

The paper is organized as follows: Section 2 discusses some related work, including document summarization and update summarization methods. Section 3 gives the problem statement and some notations that will be used throughout this paper. Section 4 reviews technical detail of PNR^2 and MRSP for comparison. Section 5 introduces our method QCQPSum. Section 6 presents the approximate method QPSum. Section 7 addresses the issue of avoiding redundancy. Section 8 shows the experimental results. Section 9 concludes the paper.

2 RELATED WORK

In this section, we review some related document summarization and update summarization methods, especially the closely related graph-ranking-based methods.

2.1 Document Summarization

Kupiec et al. [14] first apply machine learning techniques to document summarization, classifying sentences as summary sentences or nonsummary sentences. Latter work by Conroy and O'leary [6] and Shen et al. [29] is of this kind. Amini and Gallinari [2] and Wong et al. [37] explore the use of unlabeled data to improve the classifiers. Learning to rank for document summarization is also exploited by Wang et al. [35], Metzler and Kanungo [22] and Li et al. [15], which rank sentences or summaries (a set of sentences) according to their salience.

Besides the supervised methods, unsupervised methods have been applied to summarize documents as well. Nomoto and Matsumoto [27] and Radev [28] try clustering sentences and documents, respectively. The most representative sentences are selected from each cluster to form the summary. These methods aim to include different aspects of the documents in the summary. Graph-ranking-based methods are popular in recent years, which are closely related to our methods. Lexpagerank by Erkan and Radev [10] and TextRank by Mihalcea and Tarau [23], [24] are based on PageRank. A sentence graph in which a node represents a sentence is constructed in the two methods. Then, PageRank algorithm computes the scores of the sentences. Sentence selection is based on the scores. Zha [38] proposes to summarize documents through the reinforcements between words and sentences in the documents. Wan et al. [33] perform simultaneous word-to-word, word-to-sentence, and sentence-to-sentence reinforcements to determine the salience of the sentences. The rationale behind these methods is that words and sentences can reinforce each other to determine their salience. Wan et al. [32] apply Manifold Ranking algorithm to topic-focused summarization. Manifold Ranking algorithm determines the topic-biased salience of the sentences. Nastase [25] uses spreading activation technique for topic-focused summarization. Spreading activation determines the weights of the edges in the sentence graph according to a given topic. Through PageRank, topic-biased salience of the sentences is determined.

Although topic-focused summarization methods select sentences that are both salient and relevant, they cannot meet the update requirement for evolutionary documents.

2.2 Update Summarization

Wan [34] proposes the pioneering TimedTextRank algorithm to summarize evolutionary documents. TimedTextRank is adapted from the well-known graph-ranking-based method TextRank [23]. It emphasizes the importance of the current documents over the historical ones for generating novelty-oriented summaries. Technically, it gives higher weights to the current documents in the PageRank salience propagation process.

In 2007, DUC proposed the update summarization problem for evolutionary documents. There are already quite a number of approaches to attack the problem. Smmr proposed by Boudin et al. [3] extends the maximal marginal relevance method [4] for update summarization, where sentences are penalized based on their similarities to the previously read documents. The more similar a sentence to previous documents is, the more the sentence is penalized. PNR^2 by Li et al. [17] is an extension of TextRank. It performs both positive reinforcements to determine the salience of the sentences and negative reinforcements to avoid content overlapping for update summarization. It integrates previous and current documents in a unified graph for ranking. Du et al. [9] extend the manifold ranking algorithm with sink points for update summarization. The past documents are squeezed as a sink point and added into the sentence graph constructed by current documents. A modified manifold ranking algorithm is then employed to rank the sentences with sink points. With the existence of sink points, the top-ranked sentences are expected to avoid content overlapping with the previous documents, satisfying the update requirement. Wang and Li employ incremental hierarchical clustering for update summarization in [36], which can be considered as an extension of the traditional clustering-based summarization methods. Their method can summarize documents in an incremental way instead of a batch way. As soon as a new document arrives, its sentences are added into the hierarchical clustering tree. Then, the most representative sentences for the updated clusters are selected.

Some famous update summarization systems include CLASSY by Conroy et al. [8], ICSI/UTD by Gillick et al. [12], THUSUM by Chen et al. [5], etc. CLASSY 2009 uses an integer linear programming (ILP) method for redundancy removal and sentence selection. It selects a set of sentences that have the maximal sum of oracle scores, which takes into account signature terms, query terms, and the maximum likelihood estimation of the probability that a term occurs in the topic. THUSUM 2008 uses a method similar to PNR^2. With fine tuned term weights, it achieves very good ROUGE scores, which indicates term weights are quite important for the graph-ranking-based methods. ICSI/UTD 2009 has the highest ROUGE-2 score on TAC 2009. It also formulate the problem as an ILP, which seeks to select a set of sentences that maximize the sum of weights of the n-grams covered by the sentences. The system shows that sentence compression and sentence position are very important to update summarization, at least in terms of ROUGE. Some systems utilize document summarization methods for update summarization by taking a specific preprocessing or postprocessing step. For example, TAC 2009 participating ICTGrasper [39] and ICL [16] both appeal to Manifold Ranking algorithm. ICTGrasper takes a

sentence pruning preprocessing step, while ICL takes a sentence reranking postprocessing step.

3 PROBLEM STATEMENT AND NOTATIONS

A document collection is a set of documents. A document consists of a set of sentences. A topic is comprised by a set of sentences describing the information need of the users. The following is a sample topic in TAC 2008. The *narrative* expresses the information need of the users.

The problem of update summarization is formally defined as follows:

Update summarization. Given a topic T, a current document collection B and a historical document collection A, extract a set of sentences from B to compose the update summary for B supposing the users have read A.

A good update summary should comply with the main contents of B, should be relevant with the topic T, should not repeat the contents covered by A, and should not contain redundant contents. We treat the task as a sentence ranking problem. Therefore, sentences should be ranked in a way that the more relevant to the main contents of B and the topic T, and the more irrelevant to A a sentence is, the higher the sentence should be ranked. Our goal is to devise such a sentence ranking algorithm.

As for the notations, we attach to each sentence in A and B with a unique subscript. We denote a sentence, its ranking score and its topic relevance by s_i, f_i, and p_i, respectively, where i is the index of the sentence. By $s_i \in A$ (abbreviated to $i \in A$), we mean sentence s_i belongs to document collection A. The similarity between s_i and s_j is denoted by w_{ij}. Table 1 gives a summary of the notations.

4 GRAPH-RANKING-BASED METHODS

4.1 PNR^2

Ranking with Positive and Negative Reinforcement algorithm (abbr. PNR^2) [17] is an extension of the TextRank algorithm for update summarization. It constructs the following matrix:

$$\mathbf{M} = \begin{bmatrix} \alpha_1 \mathbf{W}_{AA} & \beta_1 \mathbf{W}_{AB} \\ \beta_2 \mathbf{W}_{BA} & \alpha_2 \mathbf{W}_{BB} \end{bmatrix},$$

where $\alpha_1, \alpha_2 > 0$ and $\beta_1, \beta_2 < 0$. $\alpha_1 \mathbf{W}_{AA}$ and $\alpha_2 \mathbf{W}_{BB}$ are to perform positive reinforcements. $\beta_1 \mathbf{W}_{AB}$ and $\beta_2 \mathbf{W}_{BA}$ are to perform negative reinforcements. After normalizing $\mathbf{M}$ by column, it uses the following equation to compute the ranking scores $\mathbf{f}$:

$$(\mathbf{I} - \theta \cdot \mathbf{M}) \cdot \mathbf{f} = \mathbf{p}, \tag{1}$$

where $0 < \theta < 1$.

In PNR^2, $\mathbf{M}$ is normalized in column to ensure convergence of the algorithm. With such normalization, positive reinforcements between sentences in B are biased by the sentences in A. In other words, positive reinforcements suffer from the integration of A and B. While we want to select the most salient sentences, such reinforcement process cannot achieve this end. Moreover, since users read A and B in order of time, the negative reinforcement from B to A is also unnecessary. This unnecessary negative reinforcement may lead to some undesirable effects on A and in return on B.

TABLE 1
Notations

Notations	Explanations
A, B	Sets of sentences in an evolutionary document collection at two different time stamps. A is prior to B. *We will refer to either A or B by X or Y for convenience in the following notation introduction.*
s_i	A sentence with index i
$i \in X$	Sentence s_i belongs to X.
f_i	the ranking score of s_i
$\mathbf{f}_X, \mathbf{f}$	Score vectors of the sentences in X and $A \cup B$ respectively. $\mathbf{f}_X = <f_{i_1}, f_{i_2}, ..., f_{i_n}>, i_1, ..., i_n \in X$, $\mathbf{f} = <\mathbf{f}_A^T, \mathbf{f}_B^T>^T$
p_i	the topic relevance of s_i
$\mathbf{p}_X, \mathbf{p}$	Topic relevance vectors of the sentences in X and $A \cup B$ respectively. $\mathbf{p}_X = <p_{i_1}, p_{i_2}, ..., p_{i_n}>, i_1, ..., i_n \in X$, $\mathbf{p} = <\mathbf{p}_A^T, \mathbf{p}_B^T>^T$
w_{ij}	Sentence similarity between s_i and s_j
$\mathbf{W}_{XY}$	Sentence similarity matrices between X and Y
$\mathbf{W}$	$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{AA} & \mathbf{W}_{AB} \\ \mathbf{W}_{BA} & \mathbf{W}_{BB} \end{bmatrix}$$
$diag[x_i]_{i \in X}$	Diagonal matrix with diagonal elements x_i ($i \in X$)
$\mathbf{D}_X$	$$\mathbf{D}_X = diag \left[\sum_{j \in X} w_{ij} \right]_{i \in X}$$ The ii-th element of $\mathbf{D}_X$ is denoted by D_i^X.
$\mathbf{D}_{XY}$, $(X \neq Y)$	$$\mathbf{D}_{XY} = diag \left[\sum_{j \in Y} w_{ij} \right]_{i \in X}$$ The ii-th element of $\mathbf{D}_{XY}$ is denoted by D_i^{XY}
$\mathbf{S}_X$	$\mathbf{S}_X = \mathbf{D}_X^{-1/2} \mathbf{W}_{XX} \mathbf{D}_X^{-1/2}$
$\mathbf{S}_{XY}, (X \neq Y)$	$\mathbf{S}_{XY} = \mathbf{D}_{XY}^{-1/2} \mathbf{W}_{XY} \mathbf{D}_{YX}^{-1/2}$
$\mathbf{I}$	Identity matrix

4.2 MRSP

Manifold Ranking with Sink Point algorithm [9] is an extension of the Manifold Ranking algorithm. *Sink points* are points that will absorb reinforcements spread from other points. At first, the document collection A is added into B as a single sentence, which is a sink point. MRSP uses the following iterative equation to compute the scores of the sentences:

$$\mathbf{f}_B(t+1) = \alpha \mathbf{S}_B \mathbf{I}_f \mathbf{f}_B(t) + (1-\alpha) \mathbf{p}_B, \tag{2}$$

where $\mathbf{I}_f$ is a diagonal indicator matrix with (i,i)-element equal to 0 if the ith sentence is a sink point and 1 otherwise. At every run of MRSP, the sentence having the highest score is selected and set as a sink point.

As MRSP add A into B, it will face the same issue that PNR^2 faces. That is, positive reinforcements between sentences in B are biased by the sentences in A. MRSP excludes the selected sentences from performing further reinforcements to keep the update property of the summary. However, if there are plenty of sentences that cover similar contents to the selected sentences, MRSP may fail.

In next section, we propose our graph-ranking-based method. It performs positive reinforcements in A and B, respectively, and impose the update requirement as inequality constraints. In this way, it avoids the problems that PNR^2 and MRSP suffer.

5 QUADRATICALLY CONSTRAINED QUADRATIC PROGRAMMING FORMULATION

In this section, we propose QCQPSum that formulates the update summarization problem as a quadratically constrained quadratic programming problem. We first discuss the optimization problem, then derive the algorithm.

5.1 The Optimization Problem

We want to perform reinforcements in A and B separately, which would reflect the actual salience of the sentences. Then, for update requirement consideration, we also want to impose direct suppression of A on B, suppressing the scores of B's sentences according to their similarity to A. For this purpose, we introduce a cost function w.r.t. the scores of the sentences. The cost function consists of the following terms. The first and the second terms are the regularization terms, fulfilling the reinforcements in A and B, respectively. The third term is the topic-relevance fitting term, fitting the scores to the topic relevance. The fourth term is the suppression term, fulfilling the suppression of A on B. A straightforward cost function is

$$
\begin{aligned}
\mathcal{L}(\mathbf{f}) = &\ \alpha_1 \sum_{i \in A} \sum_{j \in A} w_{ij} \left(\frac{f_i}{\sqrt{D_i^A}} - \frac{f_j}{\sqrt{D_j^A}} \right)^2 \\
&+ \alpha_2 \sum_{i \in B} \sum_{j \in B} w_{ij} \left(\frac{f_i}{\sqrt{D_i^B}} - \frac{f_j}{\sqrt{D_j^B}} \right)^2 \\
&+ \alpha_3 \sum_{i \in A \cup B} (f_i - p_i)^2 \\
&- \alpha_4 \sum_{i \in A} \sum_{j \in B} w_{ij}(f_i - f_j)^2,
\end{aligned}
$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0$.

The first, second, and the third terms are the same as the Manifold Ranking regularization framework. We give some explanation to the fourth term. From the parameter $-\alpha_4$, we can see that the greater the term is, the smaller the cost function will be. Thus, when $f_i \geq f_j$ ($i \in A$, $j \in B$), to minimize the cost function, f_i will be boosted and f_j will be suppressed. We can also see that the greater f_i is, the more f_j will be suppressed. This can be confirmed by the following simple calculation:

$$
[f_A - (f_B - \Delta f_B)]^2 = f_A^2 + (f_B - \Delta f_B)^2 - 2f_A f_B + 2f_A \Delta f_B.
\tag{3}
$$

With greater f_A, a greater decline Δf_B of f_B will lead to greater decline $2f_A \Delta f_B$ of the cost function. w_{ij} tunes the suppression according to the sentence similarity. However, there are two issues which remain to be addressed. The first is that when $f_j > f_i$ ($j \in B$, $i \in A$), A acts as boosting B instead of suppressing B. The second is that the cost

function may not have lower bounds. For the first issue, we explicitly add the following constraint:

$$
\forall w_{ij} > 0 (i \in A, j \in B), f_i \geq f_j.
\tag{4}
$$

This constraint ensures that B is suppressed by A rather than boosted by A. The more similar a sentence to A is, the more the sentence will be suppressed according to the weighting parameter w_{ij}. By introducing slack variables, the constraint becomes

$$
\forall w_{ij} > 0 (i \in A, j \in B), -f_i + f_j - \xi_{ij} \leq 0.
\tag{5}
$$

For the second issue, we constrain the solution on a hypersphere. That is, we add the following sum of square constraint:

$$
\sum_{i \in A \cup B} f_i^2 = 1.
\tag{6}
$$

Now, we formally state the optimization problem as follows:

$$
\arg \min_{\mathbf{f}, \boldsymbol{\xi}} \mathcal{L}_1(\mathbf{f}, \boldsymbol{\xi}),
\tag{7}
$$

where

$$
\begin{aligned}
\mathcal{L}_1(\mathbf{f}, \boldsymbol{\xi}) = &\ \frac{\alpha_1}{2} \sum_{i \in A} \sum_{j \in A} w_{ij} \left(\frac{f_i}{\sqrt{D_i^A}} - \frac{f_j}{\sqrt{D_j^A}} \right)^2 \\
&+ \frac{\alpha_2}{2} \sum_{i \in B} \sum_{j \in B} w_{ij} \left(\frac{f_i}{\sqrt{D_i^B}} - \frac{f_j}{\sqrt{D_j^B}} \right)^2 \\
&+ \alpha_3 \sum_{i \in A \cup B} (f_i - p_i)^2 + \beta \sum_{\substack{i \in A, j \in B \\ w_{ij} > 0}} w_{ij} \xi_{ij}^2 \\
&- \alpha_4 \sum_{i \in A} \sum_{j \in B} w_{ij}(f_i - f_j)^2,
\end{aligned}
\tag{8}
$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta > 0$, and $\beta > \alpha_4$.

Such that

$$
\forall w_{ij} > 0 (i \in A, j \in B), -f_i + f_j - \xi_{ij} \leq 0
\tag{9}
$$

$$
\sum_{i \in A \cup B} f_i^2 = 1.
\tag{10}
$$

As we will see in the next section, we divide α_1 and α_2 by 2 so that we can rewrite the cost function in a more compact matrix form. Note that we require $\beta > \alpha_4$. We will explain it later.

Now let's examine the suppression of A on B. According to the cost function, the suppression differs in two cases.

Case 1: $f_i \geq f_j$ ($i \in A, j \in B$). In this case, ξ_{ij} is zero. We suppress f_j by the term

$$
- \alpha_4 w_{ij}(f_i - f_j)^2,
\tag{11}
$$

f_i and f_j are pulled apart.

Case 2: $f_i < f_j$ ($i \in A, j \in B$). In this case, we have $\xi_{ij} = f_j - f_i$ ($i \in A, j \in B$). By substituting ξ_{ij} into $\mathcal{L}(\mathbf{f}, \boldsymbol{\xi})$, we can see that the actual suppression term is

$$
(\beta - \alpha_4) w_{ij}(f_i - f_j)^2
\tag{12}
$$

when $\beta > \alpha_4$, f_i, and f_j are pulled together. Note that this is also a kind of suppression because f_j is already greater than f_i. This also explains why we add the requirement $\beta > \alpha_4$.

From the above discussion, we can see more clearly the difference between PNR^2, MRSP, and QCQPSum. First, reinforcement within B does not involve A, and vice versa, which can be seen from the first two terms of (8). This is to make sure that we can select the most salient sentences in B. Second, A has direct suppression on B, which is obviously from the above suppression discussion. Suppression is to keep the update property of the generated summary.

Optimization problem (7) is a quadratically constrained quadratic programming problem. General optimization techniques such as Augmented Lagrangian method[4] can solve this problem. We will derive the algorithm in the next section.

5.2 Algorithm Derivation

First, we rewrite (8) in matrix form (see Table 1 for the explanation of the following notations). The first term can be rewritten as follows:

$$\frac{\alpha_1}{2}\sum_{i\in A}\sum_{j\in A} w_{ij}\left(\frac{f_i}{\sqrt{D_i^A}}-\frac{f_j}{\sqrt{D_j^A}}\right)^2$$

$$= \alpha_1\left(\sum_{i\in A}\left(\sum_{j\in A} w_{ij}\right)\frac{f_i^2}{D_i^A}-\sum_{i\in A}\sum_{j\in A} w_{ij}\frac{f_i}{\sqrt{D_i^A}}\frac{f_j}{\sqrt{D_j^A}}\right)$$

$$= \alpha_1\mathbf{f}_A^T\mathbf{D}_A^{-\frac{1}{2}}(\mathbf{I}-\mathbf{W}_{AA})\mathbf{D}_A^{-\frac{1}{2}}\mathbf{f}_A$$

$$= \alpha_1\mathbf{f}_A^T(\mathbf{I}-\mathbf{S}_{AA})\mathbf{f}_A.$$

The second term can be rewritten in the same way

$$\frac{\alpha_2}{2}\sum_{i\in B}\sum_{j\in B} w_{ij}\left(\frac{f_i}{\sqrt{D_i^B}}-\frac{f_j}{\sqrt{D_j^B}}\right)^2$$

$$= \alpha_2\mathbf{f}_B^T(\mathbf{I}-\mathbf{S}_{BB})\mathbf{f}_B.$$

The third and the fifth terms can be rewritten as follows:

$$\alpha_3\sum_{i\in A\cup B}(f_i-p_i)^2 = \alpha_3(\mathbf{f}^T\mathbf{f}-2\mathbf{p}^T\mathbf{f}+\mathbf{p}^T\mathbf{p}).$$

Since $\mathbf{p}^T\mathbf{p}$ is a constant, it can be safely discarded

$$\alpha_4\sum_{i\in A}\sum_{j\in B} w_{ij}(f_i-f_j)^2$$

$$= \alpha_4\left(\mathbf{f}_A^T\mathbf{D}_{AB}\mathbf{f}_A+\mathbf{f}_B^T\mathbf{D}_{BA}\mathbf{f}_B-2\mathbf{f}_A^T\mathbf{W}_{AB}\mathbf{f}_B\right).$$

The fourth term can be easily transformed as

$$\beta\sum_{\substack{i\in A, j\in B\\ w_{ij}>0}} w_{ij}\xi_{ij}^2 = \beta\boldsymbol{\xi}^T\mathbf{W}_\xi\boldsymbol{\xi},$$

where $\mathbf{W}_\xi$ is a diagonal matrix with diagonal elements appropriately correspond to w_{ij}.

4. The interested readers may refer to [26] for more information about augmented Lagrangian method.

Now, we define $\mathbf{g} = [\mathbf{f}^T, \boldsymbol{\xi}^T]^T$, $\mathbf{h} = [-2\alpha_3\mathbf{p}^T, \mathbf{0}^T]^T$, and

$$\mathbf{Q} = \begin{bmatrix} \gamma_1\mathbf{I}-\alpha_1 & & \\ \mathbf{S}_{AA}-\alpha_4\mathbf{D}_{AB} & \alpha_4\mathbf{W}_{AB} & \mathbf{0} \\ \alpha_4\mathbf{W}_{BA} & \gamma_2\mathbf{I}-\alpha_2\mathbf{S}_{BB}-\alpha_4\mathbf{D}_{BA} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \beta\mathbf{W}_\xi \end{bmatrix},$$

where $\gamma_1 = \alpha_1 + \alpha_3$, $\gamma_2 = \alpha_2 + \alpha_3$. The cost function in (8) can be rewritten as

$$\mathcal{L}_1(\mathbf{g}) = \mathbf{g}^T\mathbf{Q}\mathbf{g} + \mathbf{h}^T\mathbf{g}. \tag{13}$$

The two constraints (9) and (10) also can be rewritten in matrix form as

$$\mathbf{A}_i^T\mathbf{g} \leq 0 (1 \leq i \leq m) \tag{14}$$

$$\mathbf{g}^T\mathbf{B}\mathbf{g} - 1 = 0, \tag{15}$$

where $\mathbf{A}_i$ is a column vector whose elements are all zero except three elements -1, 1, and -1 at the appropriate positions and

$$\mathbf{B} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

The augmented Lagrangian function of $\mathcal{L}(\mathbf{g})$ is as follows:

$$\begin{aligned}\mathcal{L}_A^*(\mathbf{g}, \boldsymbol{\lambda}, \mu) =\ & \mathbf{g}^T\mathbf{Q}\mathbf{g} + \mathbf{h}^T\mathbf{g} \\ & + \lambda_0(\mathbf{g}^T\mathbf{B}\mathbf{g}-1) + \sum_{i=1}^m \lambda_i\left(\mathbf{A}_i^T\mathbf{g}+s_i^2\right) \\ & + \mu\left[\left(\mathbf{g}^T\mathbf{B}\mathbf{g}-1\right)^2 + \sum_{i=1}^m\left(\mathbf{A}_i^T\mathbf{g}+s_i^2\right)^2\right],\end{aligned} \tag{16}$$

where $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \dots, \lambda_m]^T$ and s_i $(1 \leq i \leq m)$ are slack variables. The above Lagrangian function can be simplified by eliminating s_i, which leads to the following equation:

$$\begin{aligned}\mathcal{L}_A(\mathbf{g}, \boldsymbol{\lambda}, \mu) =\ & \mathbf{g}^T\mathbf{Q}\mathbf{g} + \mathbf{h}^T\mathbf{g} + \lambda_0(\mathbf{g}^T\mathbf{B}\mathbf{g}-1) \\ & + \mu(\mathbf{g}^T\mathbf{B}\mathbf{g}-1)^2 - \frac{1}{4\mu}\sum_{i=1}^m \lambda_i^2 \\ & + \mu\sum_{i=1}^m\left(\max\left\{\mathbf{A}_i^T\mathbf{g}+\frac{\lambda_i}{2\mu}, 0\right\}\right)^2.\end{aligned} \tag{17}$$

When $\boldsymbol{\lambda}$ and μ are fixed, minimizing $\mathcal{L}_A(\mathbf{g}, \boldsymbol{\lambda}, \mu)$ (17) is an unconstrained problem. Gradient descent method can find a local minimum. The gradient of $\mathcal{L}_A(\mathbf{g}, \boldsymbol{\lambda}, \mu)$ w.r.t. $\mathbf{g}$ is as follows:

$$\begin{aligned}\frac{\partial\mathcal{L}_A(\mathbf{g}, \boldsymbol{\lambda}, \mu)}{\partial\mathbf{g}} =\ & 2\mathbf{Q}\mathbf{g} + \mathbf{h} + 2\lambda_0\mathbf{B}\mathbf{g} + 4\mu(\mathbf{g}^T\mathbf{B}\mathbf{g}-1)\mathbf{B}\mathbf{g} \\ & + 2\mu\sum_{i=1}^m\left(\max\left\{\mathbf{A}_i^T\mathbf{g}+\frac{\lambda_i}{2\mu}, 0\right\}\right)\mathbf{A}_i.\end{aligned} \tag{18}$$

By iteratively minimizing $\mathcal{L}_A(\mathbf{g}, \boldsymbol{\lambda}, \mu)$ w.r.t. $\mathbf{g}$ and updating $\boldsymbol{\lambda}$, μ until the solution converges, augmented Lagrangian method solves the original problem (7). A sketch of the algorithm is given by *AugmentedLagrangian*. The convergence of the algorithm can be assured without increasing μ to a very large value.

AugmentedLagrangian Augmented Lagrangian method for solving the posed regularization framework

Input:

$\mathbf{Q}, \mathbf{h}$: the quadratic cost function $\mathcal{L}(\mathbf{g}) = \mathbf{g}^T \mathbf{Q} \mathbf{g} + \mathbf{h}^T \mathbf{g}$.

$\mathbf{A}_i$: the inequality constraint $\mathbf{A}_i^T \mathbf{g} \le 0$, $1 \le i \le m$.

$\mathbf{B}$: the quadratic equality constraint $\mathbf{g}^T \mathbf{B} \mathbf{g} = 1$.

t: parameter update step length.

Output:

$\mathbf{g}$: scores of the variables.

1.   Initialize $\mathbf{g}^0$ by random assignment. Initialize λ_i
      $(i = 0, \ldots, \text{m})$ and μ to one.
2.   **for** $k = 1, 2, \ldots$
3.   Find a $\mathbf{g}^k$ that minimizes $\mathcal{L}_A(\mathbf{g}, \lambda, \mu)$ starting from
         point $\mathbf{g}^{k-1}$.
4.   **if** $\mathbf{g}^k$ converges, **then**
5.   **return** $\mathbf{g}^k$.
6.   **else**
7.   update λ, μ by
            $\lambda_0^k = \lambda_0^{k-1} + 2\mu^{k-1}(\mathbf{g}^T \mathbf{B} \mathbf{g} - 1)$,
            $\lambda_i^k = \max\{\lambda_i^{k-1} + 2\mu^{k-1}\mathbf{A}_i^T \mathbf{g}, 0\}, i = 1, \ldots, m$,
            $\mu^k = t\mu^{k-1}$.
8.   **end if**
9.   **end for**

The update summarization method for document collection B given collection A is to figure out $\mathbf{Q}, \mathbf{h}, \mathbf{A}_i$ ($1 \le i \le m$) and $\mathbf{B}$, get the ranking scores of the sentences in B through algorithm *AugmentedLagrangian*, and take the k highest scored sentences to form the summary.

5.3 Discussion

If we view scoring the sentences as putting them in a 1D space, MarginRank can be interpreted as the interaction between the points (sentences) labeled (or rather in) A and B. At first, all points are seated in their original positions (i.e., set to their prior relevance scores). Then, the interaction between the points drags them to new positions. On one hand, the points are always trying to pull neighboring points having the same label to their positions. On the other hand, the points labeled A also try to pull neighboring points labeled B in higher positions to their position and push neighboring points labeled B in lower positions away. The final positions (scores) are the results of such interaction.

QCQPsum is NP-hard. Solving QCQPSum is time consuming. In the next section, we reformulate the update summarization problem as a quadratic programming problem (QPSum), which is an approximation of QCQPSum. The cost function of the QPSum is convex, making the problem solvable in polynomial time.

6   QUADRATIC PROGRAMMING FORMULATION

In this section, we propose QPSum that formulates the update summarization problem as a convex quadratic programming problem. We first introduce the optimization problem as we did in previous section. Then, we derive the dual problem. In our experiments, the dual problem takes much less running time than the primal problem.

6.1 The Optimization Problem

We found that we could reformulate the update summarization as a quadratic programming problem by introducing

a margin ε for the inequality constraints and removing the suppression term, which leads to the following optimization problem:

$$\min_{\mathbf{f}, \boldsymbol{\xi}} \mathcal{L}_2(\mathbf{f}, \boldsymbol{\xi}) \tag{19}$$

$$
\begin{aligned}
\mathcal{L}_2(\mathbf{f}, \boldsymbol{\xi}) = & \frac{\alpha_1}{2} \sum_{i \in A} \sum_{j \in A} w_{ij} \left(\frac{f_i}{\sqrt{D_i^A}} - \frac{f_j}{\sqrt{D_j^A}} \right)^2 \\
& + \frac{\alpha_2}{2} \sum_{i \in B} \sum_{j \in B} w_{ij} \left(\frac{f_i}{\sqrt{D_i^B}} - \frac{f_j}{\sqrt{D_j^B}} \right)^2 \\
& + \alpha_3 \sum_{i \in A \cup B} (f_i - p_i)^2 + \beta \sum_{\substack{i \in A, j \in B \\ w_{ij} > 0}} w_{ij} \xi_{ij}^2,
\end{aligned}
\tag{20}
$$

where $\alpha_1, \alpha_2, \alpha_3, \beta > 0$.

Such that

$$\forall w_{ij} > 0 (i \in A, j \in B), \quad f_i - f_j \ge \varepsilon - \xi_{ij}. \tag{21}$$

This formulation has two advantages over the QCQP formulation. The first is that the cost function is convex. Following [40], the Hessian matrix of $\mathcal{L}(\mathbf{f}, \boldsymbol{\xi})$ is positive definite. Thus, $\mathcal{L}(\mathbf{f}, \boldsymbol{\xi})$ is convex. The second is that we can safely remove the hypersphere constraint on $\mathbf{f}$, which results in a convex QP problem. Compared to QCQPSum, QPSum can be solved in polynomial time. Since the solution of $\mathcal{L}(\mathbf{f}, \boldsymbol{\xi})$ obviously exists, we have the following theorem.

Theorem 1. $\mathcal{L}(\mathbf{f}, \boldsymbol{\xi})$ *has a unique global minimizer.*

Now, we compare the two problem formulations in detail. The first three terms are exactly the same, i.e., the regularization terms and the fitting term. The suppression term is different. Similar to QCQPSum, there are two cases when the suppression term is in action.

Case 1: $f_i - f_j \ge \varepsilon$ ($i \in A, j \in B$). In this case, ξ_{ij} is zero. No suppression is imposed.

Case 2: $f_i - f_j < \varepsilon$ ($i \in A, j \in B$). In this case, we have $\xi_{ij} = \varepsilon - (f_i - f_j)$ ($i \in A, j \in B$). The suppression term, i.e., suppression of s_i on s_j, is

$$\beta w_{ij}(\varepsilon - (f_i - f_j))^2. \tag{22}$$

In other words, when the difference between f_i and f_j is less than the margin ε, f_j will be suppressed by f_i. The greater the difference is, the less the cost function will be. On the contrary, when the difference is greater than the margin, the suppression is turned off. This is reasonable in the sense that if f_j is much less than f_i, further suppression does not make significant difference for the ranking results. This also explains why we say QPSum is an approximation of QCQPSum. Table 2 summarizes the suppression terms of QCQPSum and QPSum.

6.2 The Dual Problem

We first rewrite $\mathcal{L}_2(\mathbf{f}, \boldsymbol{\xi})$ in the matrix form

$$\mathcal{L}_2(\mathbf{f}, \boldsymbol{\xi}) \propto \mathbf{f}^T \mathbf{S} \mathbf{f} - 2\alpha_3 \mathbf{p}^T \mathbf{f} + \beta \boldsymbol{\xi}^T \mathbf{W}_{\xi} \boldsymbol{\xi}, \tag{23}$$

TABLE 2
Comparison of the Suppression Terms
in QCQPSum and QPSum

Formulation	Condition $(i \in A, j \in B)$	Suppression
QCQP	$f_i \geq f_j$	$-\alpha_4 w_{ij}(f_i - f_j)^2$
	$f_i < f_j$	$(\beta - \alpha_4)w_{ij}(f_i - f_j)^2$
QP	$f_i - f_j < \varepsilon$	$\beta w_{ij}(\varepsilon - (f_i - f_j))^2$
	$f_i - f_j \geq \varepsilon$	0

where

$$\mathbf{S} = \begin{bmatrix} (\alpha_1 + \alpha_3)\mathbf{I} - \alpha_1\mathbf{S}_A & 0 \\ 0 & (\alpha_2 + \alpha_3)\mathbf{I} - \alpha_2\mathbf{S}_B \end{bmatrix}. \quad (24)$$

Note that since $\alpha_3\mathbf{p}^T\mathbf{p}$ is a constant, it is removed.

Next, we rewrite the inequality constraint as follows:

$$\mathbf{Af} - \xi + \varepsilon \leq \mathbf{0}, \quad (25)$$

where the elements in each row of $\mathbf{A}$ are all 0 except two elements -1 and 1 in the appropriate positions. $\varepsilon = [\varepsilon, \ldots, \varepsilon]^T$.

The Lagrangian function of the optimization problem (19) is

$$\mathcal{L}_2(\mathbf{f}, \xi, \lambda) = \mathbf{f}^T\mathbf{S}\mathbf{f} - 2\alpha_3\mathbf{p}^T\mathbf{f} + \beta\xi^T\mathbf{W}_\xi\xi + \lambda^T(\mathbf{Af} - \xi + \varepsilon). \quad (26)$$

Setting the respective derivatives to zero, we get

$$\begin{cases} \mathbf{f} = \dfrac{1}{2}\mathbf{S}^{-1}(2\alpha_3\mathbf{p} - \mathbf{A}^T\lambda) & (27) \\ \xi = \dfrac{1}{2\beta}\mathbf{W}_\xi^{-1}\lambda. & (28) \end{cases}$$

Substituting (27) and (28) into the Lagrangian function (26), we obtain the dual problem:

$$\max_\lambda \left\{ -\frac{1}{4}\lambda^T\left(\mathbf{A}\mathbf{S}^{-1}\mathbf{A}^T + \frac{1}{\beta}\mathbf{W}_\xi^{-1}\right)\lambda + \lambda^T(\alpha_3\mathbf{A}\mathbf{S}^{-1}\mathbf{p} + \varepsilon) \right\}. \quad (29)$$

Such that

$$\lambda \geq \mathbf{0}. \quad (30)$$

Solving the dual problem is more efficient in our experiments. Note that $\mathbf{S}$ and $\mathbf{W}_\xi$ are block diagonal matrices. $\mathbf{S}^{-1}$ and $\mathbf{W}_\xi^{-1}$ can be computed by means of the inverse of each block.

6.3 Discussion

Surprisingly, we found QPSum strongly connected with the framework proposed by Agarwal [1]. In [1], the following regularization-based framework for learning a ranking function on ordered graph (abbr. OrderRank) is proposed:

$$\min_{\substack{\mathbf{f} \in \mathbb{R}^n \\ \xi \geq 0}} \left\{ \frac{1}{2}\mathbf{f}^T\mathbf{L}\mathbf{f} + C\sum_{(v_i, v_j) \in \Sigma} \tau(v_i, v_j) \cdot \xi_{ij} \right\}. \quad (31)$$

Such that

$$f_i - f_j \geq 1 - \xi_{ij} \quad \forall(v_i, v_j) \in \Sigma, \quad (32)$$

where v_i is a node. (v_i, v_j) is an edge in the graph. Σ is the set of edges. f_i is the score of v_i. $\mathbf{L}$ is the Laplacian matrix on the graph. It is defined as $\mathbf{L} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}$. C and $\tau(v_i, v_j)$ are weighting parameters for the slack variable ξ_{ij}. The constraint $f_i - f_j \geq 1 - \xi_{ij}$ is a preference constraint, meaning that v_i is preferred to v_j.

Although the inequality constraint QPSum is originally proposed for suppression, it turns out the same as the one for preference in Agarwal's regularization framework. QPSum seems exactly the same as OrderRank except for the L2 penalty and the topic-relevance fitting term. In fact, L1 penalty is also feasible in our formulation. For topic relevance, it can be transformed into a set of inequality constraints. For example, for any two sentences i and j, we can add the following constraint:

$$f_i - f_j \geq \varepsilon - \xi'_{ij}, \quad \text{if } p_i > p_j, \quad (33)$$

where ξ'_{ij} is a slack variable. Then, we remove the topic-relevance fitting term from (20) and add ξ'_{ij} into (20) by term $\beta'\sum_{i,j} \tau(p_i, p_j)\xi'_{ij}$, where $\tau(p_i, p_j)$ is a weighting parameter. Note that there are two kinds of inequality constraints. The first is the update constraint. The second is the topic-relevance constraint.

From the above discussion, we can see that the update summarization problem can be transformed into a ranking-on-ordered-graph problem. The update requirement can be satisfied by introducing a set of pairwise preferences. Thus, it is possible to use the existing approaches for ranking with pairwise preference to attack the update summarization problem.

7 OTHER ASPECTS RELATED TO THE TWO METHODS

7.1 Nonredundancy of the Summary

As we have mentioned in Section 1, a good update summary should satisfy the salience, relevance, nonredundancy, and update requirements. The two proposed methods produce summaries that meet the salience, relevance, and update requirements. In this section, we show how they can be easily adapted to satisfy the nonredundancy requirement.

Suppose we have selected a set of sentences B' to be included in the summary. Denote the remaining set of sentences as B^*. According to the nonredundancy requirement, the remaining summary sentences selected from B^* should not repeat the same contents covered by B'. It means if we rank the sentences in B^* using the proposed methods, those similar to B' should not have high ranking scores. Thus, nonredundancy requirement can be handled in the same way as the update requirement. There are two strategies to treat B'.

1. Incorporate B' into A.
2. Take B' as an independent graph.

For strategy 2, we have to modify the cost functions and the constraints a bit. For QCQPSum, the cost function (8) should be modified to be

$$\mathcal{L}_1^N(\mathbf{f}, \boldsymbol{\xi}) = \sum_{X \in \{A, B', B^*\}} \frac{\alpha_X}{2} \sum_{i \in X} \sum_{j \in X} w_{ij} \left(\frac{f_i}{\sqrt{D_i^X}} - \frac{f_j}{\sqrt{D_j^X}} \right)^2$$
$$+ \alpha_3 \sum_{i \in A \cup B' \cup B^*} (f_i - p_i)^2 + \beta \sum_{\substack{i \in A \cup B', j \in B^* \\ w_{ij} > 0}} w_{ij} \xi_{ij}^2$$
$$- \alpha_4 \sum_{i \in A \cup B'} \sum_{j \in B^*} w_{ij} (f_i - f_j)^2. \tag{34}$$

The inequality constraint (9) should be restated as follows:

$$\forall w_{ij} > 0 (i \in A \cup B', j \in B^*), -f_i + f_j - \xi_{ij} \le 0. \tag{35}$$

For QPSum, the cost function should be

$$\mathcal{L}_2^N(\mathbf{f}, \boldsymbol{\xi}) = \sum_{X \in \{A, B', B^*\}} \frac{\alpha_X}{2} \sum_{i \in X} \sum_{j \in X} w_{ij} \left(\frac{f_i}{\sqrt{D_i^X}} - \frac{f_j}{\sqrt{D_j^X}} \right)^2$$
$$+ \alpha_3 \sum_{i \in A \cup B' \cup B^*} (f_i - p_i)^2 + \beta \sum_{\substack{i \in A \cup B', j \in B^* \\ w_{ij} > 0}} w_{ij} \xi_{ij}^2. \tag{36}$$

Such that

$$\forall w_{ij} > 0 (i \in A \cup B', j \in B^*), f_i - f_j \ge \varepsilon - \xi_{ij}. \tag{37}$$

We will compare these two strategies in the experiments. From this point on, we denote the first strategy by "-I," where "I" stands for incorporated, and the second strategy by "-S," where "S" stands for separated. For example, QCQPSum-S denotes the QCQP formulation using strategy 2. An update summarization algorithm using the proposed methods is given by **UpdateSum**.

UpdateSum Update summarization algorithm
  Input:
  A, B: the snapshots of an evolutionary document
  collections at two different time stamps. The time stamp of
  A is prior to that of B
  T: the topic that expresses the information need of the
  users Tunning parameters
  Output:
  S: the summary

  1.   Set $B^* := B$, $B' := \phi$, $S := \phi$
  2.   **repeat**
  3.   Construct a unified graph of A, B^* and B'
  4.   Rank the sentences in the graph using QCQPSum
          (-I (7), (8), (9), (10), -S (7), (34), (35), (10)) or
          QPSum (-I (19), (20), (21), -S ((19), (36), (37)).
  5.   Add the sentence s_x that has the maximal score in
          B^* to S. Move s_x from B^* to B'
  6.   **if** the length of S reaches the length limit, **then**
  7.   **return** S
  8.   **end if**
  9.   **end repeat**

7.2 Normalized and Unnormalized

In the proposed QCQPSum and QPSum, the scores in the term $\sum_{i \in A} \sum_{j \in B} w_{ij}(f_i - f_j)^2$ (in (8)) and in the constraints

TABLE 3
Notations of the Variants of QCQPSum and QPSum

Notation	Description
-N	Ranking scores are normalized in the suppression terms and the inequality constraints
-U	Ranking scores are unnormalized in the suppression terms and the inequality constraints
-S	Selected sentences are treated as a separate document collection
-I	Selected sentences are incorporated into document collection A

(9), (21) can be normalized, which will result in two normalized versions. Normalizing the scores in the terms $\sum_{i \in A} \sum_{j \in B} w_{ij}(f_i - f_j)^2$ will lead to

$$\sum_{i \in A} \sum_{j \in B} w_{ij} \left(\frac{f_i}{\sqrt{D_i^{AB}}} - \frac{f_j}{\sqrt{D_j^{BA}}} \right)^2.$$

Similarly, the normalized constraints are

$$\forall w_{ij} > 0 (i \in A, j \in B), -\frac{f_i}{\sqrt{D_i^{AB}}} + \frac{f_j}{\sqrt{D_j^{BA}}} - \xi_{ij} \le 0 \tag{38}$$

for (9) and

$$\forall w_{ij} > 0 (i \in A, j \in B), \frac{f_i}{\sqrt{D_i^{AB}}} - \frac{f_j}{\sqrt{D_j^{BA}}} \ge \varepsilon - \xi_{ij} \tag{39}$$

for (21).

We will compare the normalized version with the unnormalized version in the experiments. We denote by "$-N$" and "$-U$" the normalized and unnormalized methods, respectively. For example, QCQPSum-N denotes the normalized QCQPSum method. Table 3 summarizes the notations.

8 EXPERIMENTS

8.1 Data Sets and Preprocessing

We use the popular update summarization benchmark data sets TAC 2008 and 2009[5] (TAC switched to a new summarization task called guided summarization from 2010) for our experiments. The data sets consist of a number of topics, with each associated with two document collections A and B. The automatic summarizer is expected to extract from each document collection B an update summary that does not exceed 100 words. Each document collection is associated with four summaries written by different experts as reference (ground-truth) summaries. Description of the data sets is given in Table 4.

For each document and each topic, we use the OpenNLP[6] tool to detect and tokenize sentences. A list of 707 words is used to filter stop words. The remaining words are stemmed by Snowball.[7] Then, we represent sentences and topics as word vectors, whose elements have Boolean values. We use cosine similarity to measure topic relevance and sentence similarity

5. http://www.nist.gov/tac/data/index.html.
6. http://opennlp.sourceforge.net/.
7. http://snowball.tartarus.org/index.php.

TABLE 4
Description of the Data Sets

	TAC 2008	TAC 2009
Number of Topics	48	44
Number of docs per topic	A:10 B:10	A:10 B:10
Number of reference summaries	B:4	B:4
Avg time span between A and B	109 days	82 days
Summary length	100 words	100 words

$$cos\left(\overrightarrow{s}_i, \overrightarrow{s}_j\right) = \frac{\overrightarrow{s}_i \cdot \overrightarrow{s}_j}{\|\overrightarrow{s}_i\|_2 \|\overrightarrow{s}_j\|_2}. \qquad (40)$$

8.2 Performance Evaluation

Quality of the summaries can be automatically evaluated by either ROUGE or Basic Elements (BEs),[8] which are officially adopted by TAC for evaluation of automatic generated summaries.

ROUGE. ROUGE [20] measures the performance of a summarizer by counting the overlapping units between the automatic generated summaries and the reference summaries. Given a set of reference summaries $RefSum$ and an automatic generated summary $ExtractSum$, the n-gram recall and precision measure of ROUGE are computed as follows:

$$ROUGE\text{-}N\text{-}R$$
$$= \frac{\sum_{S \in \{RefSum\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{RefSum\}} \sum_{gram_n \in S} Count(gram_n)}$$

$$ROUGE\text{-}N\text{-}P$$
$$= \frac{\sum_{S \in \{RefSum\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ExtractSum\}} \sum_{gram_n \in S} Count(gram_n)},$$

where $Count_{match}(gram_n)$ is the maximum number of n-grams $gram_n$ co-occurring in the reference summaries and the automatic generated summary. F-measure is calculated as usual:

$$ROUGE\text{-}N\text{-}F = \frac{2 \cdot ROUGE\text{-}N\text{-}R \cdot ROUGE\text{-}N\text{-}P}{ROUGE\text{-}N\text{-}R + ROUGE\text{-}N\text{-}P}.$$

Parameter setting for ROUGE is the same as the TAC official parameter setting: -n 4 -w 1.2 -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -a -d. We add another parameter "-l 100" to truncate the summaries to 100 words. According to Lin [20], ROUGE-1 and ROUGE-2 have high correlation with the human judgements. We adopt them as the evaluation measures.

Basic Elements with Transformation (BEwT). Basic Elements [31] breaks down each reference sentence into minimal semantic units, called Basic Elements. Each BE consists of a list of one to three words and their associated parts-of-speech or name-entity-recognition type. For example, all nouns, verbs, and adjectives found in the summary, subject+verb, verb+object, two head words connected via a preposition, etc. To match nonidentical units that carry the same meaning, BE evaluation method applies rules to transform each unit into a number of different variants.

---

8. ROUGE 1.5.5 package and BEwT-E package. Both can be downloaded at http://www.isi.edu/publications/licensed-sw/BE/index.html.

The evaluation procedure is as follows: The method automatically creates BEs for human summaries and machine-generated summaries, applies transformation rules to expand BE units into numerous variants, and performs matching of these units between the human summaries and machine-generated summaries. We use the build-in extraction rules and transformation rules. Tratz and Hovy [31] show that Basic Elements is a good measure. We truncate the output summaries to 100 words before the BEwT evaluation.

8.3 Experimental Results

8.3.1 Overall Performance

We set a threshold 0.3 for the similarity w_{ij} between two sentences that belong to A and B, respectively. That is, $\forall i \in A, \ j \in B$ or $i \in B, \ j \in A, \ w_{ij} = 0$ if $w_{ij} < 0.3$. We empirically tune the parameters of QCQPSum and QPSum on DUC 2007. We choose the best configuration on DUC 2007, i.e., QCQPSum-S-N and QPSum-S-U for performance comparison. For QCQPSum, there are totally five parameters $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, and β. We clench α_2 to α_1. We set α_3 to 0.05, α_4 to 0.004, β to 0.008, and α_1 to 0.938 so that they add up to 1. For QPSum, there are also five parameters $\alpha_1, \alpha_2, \alpha_3, \beta$, and ε. Again, we clench α_2 to α_1. We simple set the margin ε to 1. We set α_3 to 0.05, β to 0.0003, and α_1 to 0.9497 so that they add up to 1.

We compare the two proposed methods with the following approaches:

- *Human Avg.*: average performance of the human summarizers. The quality of summaries generated by one expert is evaluated by the ground-truth summaries generated by three other experts.
- *Baseline 1*: a summary comprised of all the leading sentences (up to 100 words) in the most recent document. Baseline 1 provides a lower bound on what can be achieved with a simple fully automatic extractive summarizer.
- *Baseline 3*: a summary consisting of sentences that have been manually selected from the document collections. Baseline 3 provides an approximate upper bound on what can be achieved with a purely extractive summarizer. It is provided by a team of five human"sentence-extractors" from the University of Montreal and available for TAC 2009 data set only.
- *Sys. Avg.*: average performance of the participating systems on TAC 2008 and 2009.
- *Best Sys.*: the best performance among all the participating systems. Note that different ROUGE-N scores may be achieved by different systems.
- *MRSP*: Manifold Ranking with Sink Point algorithm proposed in [9] recently. α in MRSP is set to 0.85 as advised in [9].
- PNR^2: graph-ranking based update summarization method proposed in [17]. The parameters are set as proposed in [17].
- *SMMR*: extension of the MMR algorithm for update summarization proposed in [3].
- *MMR baseline*: direct application of the MMR algorithm to update summarization, which computes the

TABLE 5
Performance Comparison on the TAC 2008 Data Set

	ROUGE-1	ROUGE-2	BEwT
Human Avg.	0.4003	0.11699	0.24880
Best Sys.	0.37542	0.10126	0.22337
QCQPSum-S-N	**0.37002*** [0.36313 - 0.37852]	**0.08812*** [0.08103 - 0.09522]	**0.20143**
QPSum-S-U	0.3667* [0.35875 - 0.37543]	0.0858* [0.07913 - 0.09279]	0.20065
MRSP	0.33219	0.07452	0.16917
PNR^2	0.3259	0.06994	0.15739
SMMR	0.31936	0.06921	0.15412
MMR baseline	0.31875	0.06531	0.15238
Relevance (1)	0.31961	0.06772	0.15425
Relevance (0.5)	0.31999	0.06947	0.15964
Relevance (0.3)	0.29078	0.04547	0.11886
Sys. Avg.	0.32151	0.068	0.16542
Baseline 1	0.28149	0.05975	0.14384

All ROUGE scores refer to ROUGE-N-R.

similarities between the sentences in B and the summary sentences in A.

- *Relevance (N)*: an update summarization method that rank sentences according to their relevance (cosine similarity) to the topic in B. It first removes the set of sentences $S = \{s_i \in B | \max_{s_j \in A} sim(s_i, s_j) > N\}$ from B as a preprocessing step.

Tables 5 and 6 show the performance of QCQPSum-S-U, QPSum-S-U and the comparative approaches. The performance of the comparative algorithms is based on our understanding and reimplementation.

We can see that QCQPSum and QPSum significantly outperform the comparative automatic approaches at a confidence level of 95 percent and that QPSum achieves comparable performance to QCQPSum. Through the approach Relevance, we can see that a simple sentence prepruning approach does not work well. It is interesting to see that QPSum-S-U outperforms QCQPSum-S-N in ROUGE-2 recall for the former means to be an approximation of the latter. Since their performance are so close, we believe it only has something to do with the parameter setting.

QCQPSum ranks fifth and eighth in ROUGE-1-R among all the systems on TAC 2008 and 2009, respectively, while QPSum ranks sixth and eighth. Note that the best systems

TABLE 6
Performance Comparison on the TAC 2009 Data Set

	ROUGE-1	ROUGE-2	BEwT
Human Avg.	0.39416	0.10677	0.23681
Baseline 3	0.37044	0.09799	0.21521
Best Sys.	37646	0.104	0.21727
QCQPSum-S-N	**0.36247*** [0.35501 - 0.36997]	0.08554* [0.08202 - 0.09078]	**0.19255**
QPSum-S-U	0.36197* [0.35462 - 0.36958]	**0.0861*** [0.08075 - 0.0915]	0.18816
MRSP	0.34928	0.07613	0.17288
PNR^2	0.33934	0.06971	0.15937
SMMR	0.34073	0.07322	0.16267
MMR baseline	0.33931	0.07206	0.16102
Relevance (1)	0.33773	0.07275	0.16612
Relevance (0.5)	0.341	0.07519	0.16756
Relevance (0.3)	0.30843	0.05468	0.13693
Sys. Avg.	0.33452	0.07647	0.16642
Baseline 1	0.28591	0.05115	0.131124

*All ROUGE scores refer to ROUGE-N-R. * indicates that the improvement over all the comparative automatic summarization algorithms is statistically significant at a confidence level of 95 percent.*

TABLE 7
Effects of the Suppression Terms in QCQPSum and QPSum

	TAC 2008		TAC 2009	
	off	on	off	on
QCQPSum-S-N	0.08064	0.08812	0.07925	0.08554
QPSum-S-U	0.07915	0.0858	0.08003	0.0861

All scores refer to ROUGE-2-R.

use many optimizing techniques to obtain better results and ROUGE scores while our methods use only standard text preprocessing and no further postprocessing. Our methods outperform the core algorithms of the systems that achieve the highest ROUGE scores on TAC 2008 and 2009. For example, the core algorithm of the system THUSUM [5], which achieves the highest ROUGE-1 on TAC 2008, is similar to PNR^2. Our methods significantly outperform PNR^2. The core algorithm of the system ICSI_UTD [12], which has the highest ROUGE-1-R on TAC 2009, is ICSI 2008 [11]. It achieves 0.359 in ROUGE-1-R on TAC 2008. We can see our methods also outperform ICSI 2008 in ROUGE-1-R. We have tried a simple heuristic—upweighting the beginning sentence in every document because the first sentence is usually more important [12]. Though simple, it greatly improves the performance of our methods. The ROUGE-2 recall of QCQPSum improves from 0.08812 to 0.09653 on TAC 2008 and from 0.08554 to 0.08982 on TAC 2009. The improvement on TAC 2008 is statistically significant. The ROUGE-2 recall of QPSum also improves from 0.0858 to 0.09512 on TAC 2008 and from 0.0861 to 0.09205 on TAC 2009. The improvement on both data sets is statistically significant. We believe that some task-specific processing is quite effective in improving the performance of a summarization system. We leave it to future work to incorporate more task-specific processing with our methods.

To examine the effects of the suppression terms in QCQPSum and QPSum, we deliberately turn the suppression terms off, i.e., setting the corresponding coefficients to zero, to see the resulting performance. For QCQPSum, we set α_4 and β to 0. For QPSum, we set β to 0. The other parameters are left the same as described before. We refer to this parameter setting as "off" and the one described previously as "on." Table 7 shows the ROUGE-2-R scores of both algorithms in both parameter settings. We can see that the suppression terms do greatly enhance the performance of the algorithms.

8.3.2 Comparison of Different Strategies

As we have mentioned in Section 7, there are two strategies to treat the selected sentences, which leads to QCQPSum-S, QPSum-S, QCQPSum-I, and QPSum-I. Also, the scores in the suppression terms can be either normalized or unnormalized, which leads to QCQPSum-N, QPSum-N, QCQPSum-U, and QPSum-U. We now compare different combinations of these strategies. Tables 8 and 9 show the experimental results. We can see that in fact the performance of QCQPSum and QPSum does not differ much with different strategies.

TABLE 8
Comparison of Different Strategies on TAC 2008

	ROUGE-1	ROUGE-2
QCQPSum-S-U	0.36918	0.08701
QCQPSum-I-U	0.36772	0.08709
QCQPSum-S-N	0.37002	0.08812
QCQPSum-I-N	0.3655	0.08689
QPSum-S-U	0.3667	0.0858
QPSum-I-U	0.3667	0.0858
QPSum-S-N	0.36472	0.08473
QPSum-I-N	0.36468	0.08481

All ROUGE scores refer to ROUGE-N-R.

TABLE 9
Comparison of Different Strategies on TAC 2009

	ROUGE-1	ROUGE-2
QCQPSum-S-U	0.36133	0.08549
QCQPSum-I-U	0.36019	0.0848
QCQPSum-S-N	0.36247	0.08554
QCQPSum-I-N	0.36129	0.08446
QPSum-S-U	0.36197	0.0861
QPSum-I-U	0.36054	0.08483
QPSum-S-N	0.36073	0.08478
QPSum-I-N	0.36033	0.08443

All ROUGE scores refer to ROUGE-N-R.
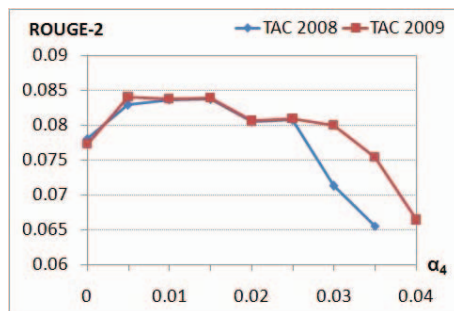
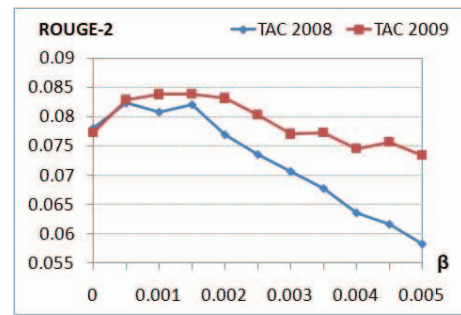Fig. 1. ROUGE-2-F of QCQPSum w.r.t. α_4. β is simply set to $2\alpha_4$.

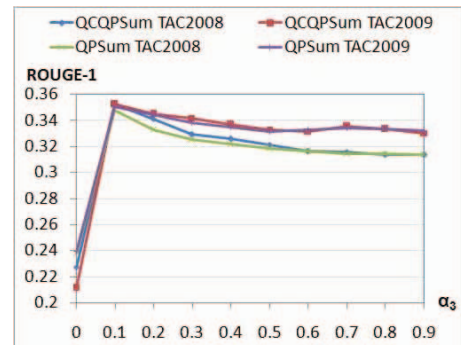Fig. 2. ROUGE-2-F of QPSum w.r.t. β.

Fig. 3. ROUGE-1-F of QPSum w.r.t. α_3.
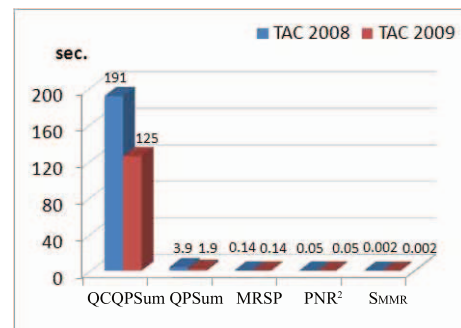
Fig. 4. Running time of the algorithms. The running time is measured in second per document collection.

8.3.3 Parameter Tuning

In this section, we tune the parameters α_4, β, and α_3 to exam the effects of the suppression terms and the topic-relevance terms in QCQPSum and QPSum.

Since the suppression term plays an important role for update summarization, we vary its coefficients to see its effects. First, we do experiments with QCQPSum. According to the analysis in Section 5.1 ((11) and (12)), coefficients of the suppression terms are actually $-\alpha_4$ and $\beta - \alpha_4$. We simply set β to $2\alpha_4$. We fix α_1, α_2 to 0.95, α_3 to 0.05. Then, we gradually increase α_4. Fig. 1 shows the experimental results.

We can see that the performance of QCQPSum approximately improves by 0.007 in ROUGE-2 on TAC 2008 and 2009 data sets. This is statistically significant improvement. It shows that suppressing the scores of the sentences with little information novelty helps to generate better summaries.

Next, we do experiments with QPSum, which is to vary β (20). Again, we fix α_1, α_2 to 0.95, α_3 to 0.05, and gradually increase β. Fig. 2 shows the experimental results. QPSum also can be improved by approximately 0.007 in ROUGE-2 with the an appropriately tuned β.

We tune α_3 to see the effects of the topic relevance term. We fix α_4 to 0.004, β to 0.008 for QCQPSum, and β to 0.0003 for QPSum. α_1 and α_2 is set to $1 - \alpha_3$ for both QCQPSum

and QPSum. We vary α_3 from 0 to 0.9. Fig. 3 shows the results. When setting α_3 to 0, the performance of both QCQPSum and QPSum is very poor. Because both approaches tend to select sentences irrelevant to the topic in this circumstance. It seems that α_3 should be set to a very small value.

8.3.4 Running Time

Summarization methods should be scalable to deal with large amount of information. We compare the running time of QCQPSum and QPSum with several other summarization methods. The time to summarize a single document collection is shown in Fig. 4. The experiments are conducted on a desktop computer with an Intel Core2 Quad 2.66 GHz CPU and 3.37 GB memories. The operating system is Windows XP sp3. The programming language is Matlab.

We can see that QCQPSum is relatively slow compared to the other methods while QPSum achieves comparable efficiency to the other methods. So, QPSum may be a better choice for update summarization due to its effectiveness and efficiency.

9 CONCLUSION AND FUTURE WORK

In this paper, we propose a new graph-ranking-based method QCQPSum for update summarization. It formulates the task as a QCQP problem. According to the problem formulation, previous document does not directly participate in the reinforcement propagation in current documents but acts as constraints. Thus QCQPSum avoids the problem in PNR^2 and MRSP, i.e., salience determination of the sentences in current document will be disturbed by previous documents. Since QCQPSum is NP-hard, we propose its approximate method QPSum, which formulates the update summarization problem as a QP problem. QPSum is polynomial time solvable. Experiments show that it outperforms the other graph-ranking-based approaches and achieves comparable efficiency.

Some NLP processing and task-specific techniques are very helpful for summarization. We already show that upweighting the first sentence in a document can greatly improve the performance of our method. As future work, we will seek more NLP processing and task-specific techniques to further improve the performance of our approach.

REFERENCES

[1] S. Agarwal, "Ranking on Graph Data," *Proc. Int'l Conf. Machine Learning (ICML '06)*, pp. 25-32, 2006.

[2] M.-R. Amini and P. Gallinari, "The Use of Unlabeled Data to Improve Supervised Learning for Text Summarization," *Proc. 25th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '02)*, pp. 105-112, 2002.

[3] F. Boudin, M. El-Bèze, and J.-M. Torres-Moreno, "A Scalable MMR Approach to Sentence Scoring for Multi-Document Update Summarization," *Proc. Int'l Conf. Computational Linguistics (COLING '08)*, pp. 23-26, 2008.

[4] J. Carbonell and J. Goldstein, "The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries," *Proc. 21st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '98)*, pp. 335-336, 1998.

[5] S. Chen, Y. Yu, C. Long, F. Jin, L. Qin, M. Huang, and Z. Xiaoyan, "Tsinghua University at the Summarization Track of TAC 2008," *Proc. Text Analysis Conf. (TAC '08)*, 2008.

[6] J.M. Conroy and D.P. O'leary, "Text Summarization via Hidden Markov Models," *Proc. 24th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '01)*, pp. 406-407, 2001.

[7] J.M. Conroy and J.D. Schlesinger, "CLASSY 2007 at DUC 2007," *Proc. Document Understanding Conf. (DUC '07)*, 2007.

[8] J.M. Conroy, J.D. Schlesinger, and D.P. O'leary, "CLASSY 2009: Summarization and Metrics," *Proc. Text Analysis Conf. (TAC '09)*, 2009.

[9] P. Du, J. Guo, J. Zhang, and X. Cheng, "Manifold Ranking with Sink Points for Update Summarization," *Proc. ACM Int'l Conf. Information and Knowledge Management (CIKM '10)*, pp. 1757-1760, 2010.

[10] G. Erkan and D.R. Radev, "Lexpagerank: Prestige in Multi-Document Text Summarization," *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP '04)*, 2004.

[11] D. Gillick, B. Favre, and D. Hakkani-Tür, "The ICSI Summarization System at TAC 2008," *Proc. Text Analysis Conf. (TAC '08)*, 2008.

[12] D. Gillick, B. Favre, D. Hakkani-Tür, B. Bohnet, Y. Liu, and S. Xie, "The ICSI/UTD Summarization System at TAC 2009," *Proc. Text Analysis Conf. (TAC '09)*, 2009.

[13] K. Knight and D. Marcu, "Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression," *Artificial Intelligence*, vol. 139, no. 1, pp. 91-107, 2002.

[14] J. Kupiec, J. Pedersen, and F. Chen, "A Trainable Document Summarizer," *Proc. 18th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '95)*, pp. 68-73, 1995.

[15] L.-D. Li, K. Zhou, G.-R. Xue, H.-Y. Zha, and Y. Yu, "Enhancing Diversity, Coverage and Balance for Summarization through Structure Learning," *Proc. 18th Int'l Conf. World Wide Web (WWW '09)*, pp. 71-80, 2009.

[16] S.-J. Li, W. Wang, and Y.-W. Zhang, "TAC 2009 Update Summarization of ICL," *Proc. Text Analysis Conf. (TAC '09)*, 2009.

[17] W. Li, F. Wei, Q. Lu, and Y. He, "PNR^2: Ranking Sentences with Positive and Negative Reinforcement For Query-Oriented Update Summarization," *Proc. 22nd Int'l Conf. Computational Linguistics (COLING '08)*, pp. 489-496, 2008.

[18] X. Li, Y.-D. Shen, L. Du, and C.-Y. Xiong, "Exploiting Novelty, Coverage and Balance for Topic-Focused Multi-Document Summarization," *Proc. ACM Int'l Conf. Information and Knowledge Management (CIKM '10)*, pp. 1765-1768, 2010.

[19] X. Li, L. Du, and Y.-D. Shen, "Graph-Based Marginal Ranking for Update Summarization," *Proc. SIAM Int'l Conf. Data Mining (SDM '11)*, pp. 486-497, 2011.

[20] C.-Y. Lin, "Rouge: A Package for Automatic Evaluation of Summaries." *Proc. ACL Workshop Text Summarization Branches Out*, p. 10, 2004.

[21] H.P. Luhn, "Automatic Creation of Literature Abstracts," *IBM J. Research Development*, vol. 2, pp. 159-165, 1958.

[22] D. Metzler and T. Kanungo, "Machine Learned Sentence Selection Strategies for Query-Biased Summarization." *Proc. SIGIR Learning to Rank Workshop*, 2008.

[23] R. Mihalcea and P. Tarau, "TextRank: Bringing Order Into Texts," *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP '04)*, 2004.

[24] R. Mihalcea, "Language Independent Extractive Summarization," *Proc. ACL Interactive Poster and Demonstration Sessions (ACL '05)*, pp. 49-52, 2005.

[25] V. Nastase, "Topic-Driven Multi-Document Summarization with Encyclopedic Knowledge and Spreading Activation." *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP '08)*, pp. 763-772, 2008.

[26] J. Nocedal and S.J. Wright, *Numerical Optimization.* Springer-Verlag, 1999.

[27] T. Nomoto and Y. Matsumoto, "A New Approach to Unsupervised Text Summarization," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '01)*, pp. 26-34, 2001.

[28] D.R. Radev, H.-Y. Jing, M. Styś, and D. Tam, "Centroid-Based Summarization of Multiple Documents," *Information Processing and Management*, vol. 40, no. 6, pp. 919-938, 2004.

[29] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document Summarization Using Conditional Random Fields," *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI '07)*, pp. 2862-2867, 2007.

[30] A. Tombros and M. Sanderson, "Advantages of Query Biased Summaries in Information Retrieval," *Proc. 21st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '98)*, pp. 2-10, 1998.

[31] S. Tratz and E. Hovy, "Summarization Evaluation Using Transformed Basic Elements," *Proc. Text Analytics Conf.*, 2008.

[32] X. Wan, J. Yang, and J. Xiao, "Manifold-Ranking Based Topic-Focused Multi-Document Summarization," *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI '07)*, pp. 2903-2908, 2007.

[33] X. Wan, J. Yang, and J. Xiao, "Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction," *Proc. Ann. Meeting Assoc. Computational Linguistics (ACL '07)*, pp. 552-559, 2007.

[34] X. Wan, "TimedTextRank: Adding the Temporal Dimension to Multi-Document Summarization," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '07)*, pp. 867-868, 2007.

[35] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang, "Learning Query-Biased Web Page Summarization," *Proc. Conf. Information and Knowledge Management (CIKM '07)*, pp. 555-562, 2007.

[36] D. Wang and T. Li, "Document Update Summarization Using Incremental Hierarchical Clustering," *Proc. Conf. Information and Knowledge Management (CIKM '10)*, pp. 279-288, 2010.

[37] K.-F. Wong, M. Wu, and W. Li, "Extractive Summarization Using Supervised and Semi-Supervised Learning," *Proc. 22nd Int'l Conf. Computational Linguistics (COLING '08)*, pp. 985-992, 2008.

[38] H.-Y. Zha, "Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '02)*, pp. 113-120, 2002.

[39] J. Zhang, P. Du, H.-B. Xu, and X.-Q. Cheng, "ICTGrasper at TAC2009: Temporal Preferred Update Summarization." *Proc. Text Analysis Conf. (TAC '09)*, 2009.

[40] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on Data Manifolds," *Proc. Advances in Neural Information Processing Systems (NIPS '03)*, pp. 169-176, 2003.

Liang Du received the BS degree in software engineering from Wuhan University, China, in 2007. He is currently working toward the PhD degree in computer science at the Institute of Software, Chinese Academy of Sciences. His research interests include machine learning, data mining and information retrieval.

Yi-Dong Shen is a professor of computer science in the State Key Laboratory of Computer Science at the Institute of Software, the Chinese Academy of Sciences, China. Prior to joining this laboratory, he was a professor at Chongqing University, China. His main research interests include knowledge representation and reasoning, semantic web, and data mining.

Xuan Li received the BS degree in computer science and technology from Sun Yat-Sen University, China, in 2006. He is currently working toward the PhD degree in computer science at the Institute of Software, Chinese Academy of Sciences. His research interests include machine learning, data mining and document summarization.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.