2011 11th IEEE International Conference on Data Mining

Learning to Rank for Query-focused Multi-Document Summarization

Chao Shen, Tao Li
School of Computing and Information Sciences
Florida International University, Miami, Florida 33199
Email: {cshen001|taoli}@cs.fiu.edu

***Abstract*—In this paper, we explore how to use ranking SVM to train the feature weights for query-focused multi-document summarization. To apply a supervised learning method to sentence extraction in multi-document summarization, we need to derive the sentence labels for training corpus from the existing human labeling data in form of <query, document set, human summaries>. However, this process is not trivial, because the human summaries are abstractive, and do not necessarily well match the sentences in the documents. In this paper, we try to address the above problem from the following two aspects. First, we make use of sentence-to-sentence relationships to better estimate the probability of a sentence in the document set to be a summary sentence. Second, to make the derived training data less sensitive, we adopt a cost sensitive loss in the ranking SVM's objective function. The experimental results demonstrate the effectiveness of our proposed method.**

***Keywords*-query-based multi-document summarization; learning to rank**

I. Introduction

As a fundamental and effective tool for document understanding, organization, and navigation, query-focused multi-document summarization has been very active and enjoying a growing amount of attention with the ever-increasing growth of the on-line document data (e.g., news, emails, blogs, web pages). For query-focused multi-document summarization, a summarizer incorporates user declared queries and generates summaries that not only reflect the important concepts in the input documents but also bias to the queries. Query-focused multi-document summarization methods can be broadly classified into two types: extractive summarization and abstractive summarization. Extractive summarization usually selects phrases or sentences from the input documents while abstractive summarization involves paraphrasing components of input documents and sentence reformulation [1].

There are many recent studies on query-focused multi-document summarization and most proposed techniques are extractive methods. Typical examples include methods based on knowledge in Wikipedia [2], information distance [3], non-negative matrix factorization [4], graph theory [5] and graph ranking [6; 7].

Generally speaking, the extracted sentences in the summary should be *representative* or *salient*, capturing the important content related to the queries with *minimal redundancy* [8]. In particular, these extractive summarization methods typically select the sentences in the input documents to form the summary based on a set of content or linguistic features, such as term frequency-inverse sentence frequency (tf-isf), sentence or term position, salient or informative keywords, and discourse information. Various features have been used to characterize the different aspects of the sentences and their relevance to the queries.

A. Supervised Learning for Summarization

With the accumulation of human summaries generated for summarization evaluation such as those used in Document Understanding Conference (DUC)[1] and the development of various features indicating the saliency/importance of the sentences, it is thus possible to explore the problem of how to calculate the combinational effects of various features and perform sentence extraction using *supervised learning* methods.

Note that most existing human labeling data of query-focused multi-document summarization is in the form of triples <query, document set, human summaries>. In order to make use of this kind of data, and apply a standard supervised learning algorithm (classification/regression/ranking) to learn a model to rank the sentences for a new <query, document set> pair, the existing human labeling data needs to be transformed first to generate the training data for supervised learning, that is, to assign a label/score for each sentence. The general framework of an extractive summarization system using supervised learning is given in Figure 1. The framework consists of the following major components: (1) training data generation where the given human summaries are transformed into the training data for supervised learning; (2) model learning where a supervised learning model is constructed to label/rank the sentences; and (3) summary generation for new documents where the learned model is used for ranking the sentences followed by redundancy removal. Note the data transformation is not trivial, because human-generated summaries are abstractive and do not necessarily well match the sentences in the documents. To solve this problem, in this paper, both the training data generation and the subsequent model learning component are considered.

[1] http://duc.nist.gov

1550-4786/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDM.2011.91

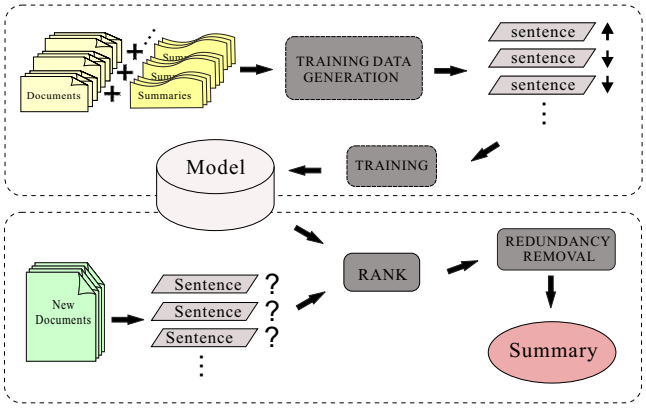


Figure 1. The framework of supervised learning for summarization.

B. Contributions of the Paper

Recently, support vector regression (SVR), has been used to automatically combine various sentence features for supervised summarization [9]. However, since we only need to differentiate the "summary sentence" and "non-summary sentence", the model is not necessary to fit the regression scores of the training data. In other words, it should make no difference if we swap two non-summary sentences which are ranked low in a ranked sentence list, even thougth their regression scores are different. So the objective in regression model learning is too aggressive, measuring the average distance between the predicted score and the true score for all sentences. Another reason of the problem of regression model is that the true score for a sentence in the training set is estimated automatically and the quality of the estimation is not guaranteed.

In this paper, we propose a method for text summarization based on ranking techniques and explore the use of ranking SVM [10], a learning to rank method, to train the feature weights for query-focused multi-document summarization. To construct the training data for ranking SVM, a rank label of "summary sentence" or "non-summary sentence" needs to be assigned to the training sentences. This assignment generally relies on a threshold of sentence scoring. Our experiments show that a small variation of the threshold may lead to a substantial change on the performance of the trained model. The sentences near the threshold are likely to be assigned with a wrong rank label, thus, introducing noise into the training set. To make the threshold less sensitive, we adopt a cost sensitive loss in the ranking SVM's objective function, giving less weights to those sentence pairs whose relative positions are of less certainty. While there are existing works on using ranking for summarization, the proposed method of cost sensitive loss will improve the robustness of learning and extend the usefulness of rank-based summarization techniques.

Our work also contributes to training data generation for supervised summarization. Note that the problem of automatic training data generation is essential in trainable summarizers. To better estimate the probability of a sentence in the document set to be a summary sentence, we propose a novel method by utilizing the sentence relationships to improve the estimation of the probability in training data generation.

The rest of the paper is organized as follows. In Section II, we review the related work about supervised learning for summarization and learning to rank. Then we discuss in detail the ranking SVM and how to adapt the ranking SVM for summarization by applying cost sensitive loss in Section III. In Section IV, we discuss how we improve the training data generation by utilizing the sentence relationships. Features used in this work are listed in Section V. Section VI presents the experimental results and analysis, and finally Section VII concludes the paper.

II. Related Work

A. Supervised Learning for Summarization

Supervised learning approaches have been successfully applied in single document summarization, where the training data is available or easy to build. The most straightforward way is to regard the sentence extraction task as a binary classification problem. Kupiec et al [11] developed a trainable summarization system which adopted various features and used a Bayesian classifier to learn the feature weights. The system performed better than other systems using only a single feature. Hirao et al [12] trained a SVM model for important sentence extraction and the model outperformed other classification models such as decision-tree or boosting methods on the Japanese Text Summarization Challenge (TSC). To make use of the sentence relations in a single document, sequential labeling methods are used to extract a summary for a single document. Zhou and Hovy [13] applied a HMM-based model and Shen et al [14] proposed a conditional random field based framework.

For query-focused multi-document summarization, Zhao et al [15] applied the Conditional Maximum Entropy, a classification model, on the DUC 2005 query-based summarization task. Similar to those methods developed for single document summarization, the model was trained on an existing training dataset where sentences are labeled as summary or non-summary manually. Ouyang et al [9] constructed the training data by labeling the sentence with a "true" score calculated according to human summaries, and then used support vector regression (SVR) to relate the "true" score of the sentence to its features. Similar to [9], in this paper, we construct the training data from human summaries. However, the learning to rank method is used in our work for query-focused multi-document summarization.

B. Learning to Rank

Learning to rank, in parallel with learning for classification and regression, has been attracting increasing interests in statistical learning for the last decade, because many applications such as web search and retrieval can be formalized as ranking problems.

Many of the learning to rank approaches are pairwise approaches, where the learning to rank problem is approximated by a classification problem, and a classifier is learned to tell whether a document is better than another. Recently, a number of authors have proposed directly defining a loss function on a list of objects and directly optimizing the loss function in learning [16; 17]. Most of these list-wise approaches directly optimize a performance measure in information retrieval, such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [18].

In the summarization task, there is no clear performance measure for the ranked sentence list. Note that the ranked sentence list is still an intermediate result for summarization and redundancy removal is needed to form the final summary. Hence, we develop our summarization system based on ranking SVM, a typical pairwise learning to rank method. Other pairwise learning to rank methods include RankBoost [19] and RankNet [20]. Our modification of ranking SVM is inspired by adopting cost sensitive loss function to differentiate document pairs from different queries or in different ranks [21; 22].

Most learning to rank methods, however, are based on the available high-quality training data. This is not the case when we apply these methods for summarization, where the training data needs to be automatically generated from the set of <query, document set, human summaries> triples.

III. Model Learning

Under the feature-based summarization framework, normally the scoring function needs to combine the impacts of various features. A common way is to use the linear combination of the features by tuning the weights of the features manually or empirically. The problem of such a method is that when the number of the features gets larger, the complexity of assigning weights grows exponentially. In this section, we explore the use of ranking SVM, a pairwise learning to rank model, for obtaining credible and controllable solutions for feature combinations.

A. Ranking SVM

Assume that a training set of labeled data is available. Given a training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ with $\mathbf{x}_i \in \Re^N$ and $y_i \in \{1, \ldots, R\}$. In the formulation of Herbrich et al. [23], the goal is to learn a function $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, so that for any pair of examples $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ it holds that

$$h(\mathbf{x}_i) > h(\mathbf{x}_j) \Longleftrightarrow y_i > y_j.$$

Table I
EXAMPLE RANKINGS FOR THE FIVE SENTENCES.

	@1	@2	@3	@4	@5
Ranking 1(Perfect) :	s(.3)	s(.7)	n(.3)	n(.7)	n(.8)
Ranking 2(Perfect) :	s(.7)	s(.3)	n(.7)	n(.3)	n(.8)
Ranking 3 :	s(.3)	n(.7)	s(.7)	n(.3)	n(.8)
Ranking 4 :	s(.7)	n(.3)	s(.3)	n(.7)	n(.8)

In this way, the task of learning to rank is formulated as the problem of classification on pairs of instances. In particular, the SVM model can be applied and the task is thus formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, \xi_{ij} \geq 0} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + \tfrac{C}{m} \sum_{(i,j) \in P} \xi_{ij} \\ s.t. \quad & \forall (i,j) \in P : \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j) > 1 - \xi_{ij}, \end{aligned} \tag{1}$$

where P is the set of pairs (i, j) for which example i has a higher rank than example j, i.e. $P = \{(i, j) : y_i > y_j\}$, $m = |P|$, and ξ_{ij}'s are slack variables. This optimization problem is equivalent to

$$\min_{\mathbf{w}} \frac{1}{2C}\mathbf{w}^T\mathbf{w} + \frac{1}{m} \sum_{(i,j) \in P} \max\{0, 1 - \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)\}, \tag{2}$$

where the second term is called "empirical hinge loss".

B. Cost Sensitive Loss

Since the rankings of the sentences in the training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are estimated, applying the empirical hinge loss may not be proper. Let us consider the following example. Given five sentences $\{s(.3), s(.7), n(.3), n(.7), n(.8)\}$, where 's' and 'n' indicate two possible ranks: summary and non-summary respectively, and the value in parentheses indicates the confidence score of the rank. Table I shows four possible rankings for these five sentences. Ranking 1 and Ranking 2 are both perfect, since it does not matter to swap two positions of both non-summary sentences or both summary sentences. Apparently, neither Ranking 3 nor Ranking 4 is perfect, and without considering confidence, they have the same quality. However, Ranking 4 should be better than Ranking 3 if we take the confidence into consideration. For the pair $< n(.7), s(.7) >$ in Ranking 3, $n(.7)$ is likely to be a non-summary sentence, and $s(.7)$ is likely to be a summary sentence. Therefore, we have good confidence that their relative positions should be swapped. For the pair $< n(.3), s(.3) >$ in Ranking 4, $n(.3)$ is less likely to be a non-summary sentence and $s(.3)$ is less likely to be a summary sentence. Their relative positions may be correct while their ranks might be mislabeled.

To deal with this problem, we adopt the idea of sensitive cost loss for SVM, and use penalty weight σ_{ij} for the loss function of each sentence pair. So the optimization problem in Eq. (2) becomes

$$\min_{\mathbf{w}} \frac{1}{2C}\mathbf{w}^T\mathbf{w} + \frac{1}{m}\sum_{(i,j)\in P} \max\{0, \sigma_{ij}(1 - \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j))\}. \tag{3}$$

In our task, for the sentence pair $< \mathbf{x}_i, \mathbf{x}_j >$, the sum of confidence scores of $\mathbf{x}_i$ and $\mathbf{x}_j$ (represented by c_i and c_j, respectively) can be used as the penalty weights. In other words,

$$\sigma_{ij} = c_i + c_j.$$

Basically, a pair of a non-summary sentence and a summary sentence with small confidence having reversed relative ranking positions will be less penalized than those with high confidence. To solve the problem in Eq.(3), we can solve the equivalent problem

$$\begin{array}{ll} \min\limits_{\mathbf{w},\xi_{ij}\geq 0} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{m}\sum\limits_{(i,j)\in P}\xi_{ij} \\ s.t. & \forall (i,j) \in P : \sigma_{ij}(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)) \geq \sigma_{ij} - \xi_{ij}. \end{array} \tag{4}$$

IV. Training Data Construction: A Graph based Method

In order to apply learning to rank for summarization, we need to have the labeled training set in the form of $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i$ is a sentence and y_i is the ranking of the sentence. Given a set of triples <query, document set, manual summary set>, instead of manually labeling the rank for every sentence, which is a time-consuming task, we can estimate the rank of a sentence with the reference of the manual summaries. For simplicity, we only assign the sentence with two possible ranks: summary or non-summary[2].

Note that generally human summaries do not contain redundancy. Therefore, to construct the training data, the sentences that have similar meanings to sentences in human summaries but of lexical diversity should also be labeled as summary sentences. So, instead of simply comparing sentences in the document set with those in human summaries, we take the training data construction as an extractive summarization task, where the similarity between sentences in the documents set are also considered, and similar sentences should have similar probabilities to be labeled as a summary sentence [6]. Different from a standard extractive summarization, here redundancy removal is performed and the human summaries are used as the query.

To estimate the probability score $p(s|H)$ of a sentence s being labeled as a summary sentence given the human summary set H, we measure its relevance with sentences in the human summary set and its similarities with the other sentences in the document set. Formally, $p(s|H)$ is computed by the following formula:

$$\begin{array}{rl} p(s|H) = & d\sum\limits_{v\in C}\frac{sim(s,v)}{\sum_{z\in C} sim(z,v)}p(v|H) \\ & +(1-d)\frac{rel(s,H)}{\sum_{z\in C} rel(z,H)}, \end{array} \tag{5}$$

where C is the set of all sentences in the document set, and d is a trade-off parameter in the interval [0, 1], used to specify the relative contribution of the two terms in Eq.(5). For bigger value of d, more importance is given to the sentence-to-sentence similarity compared to sentence-to-human-summary relevance. The denominators in both terms are used for normalization. The matrix form of Eq.(5) can be written as

$$p(k+1) = M^T p(k), \tag{6}$$

$$M = dA + (1-d)B, \tag{7}$$

where M, A, and B are all square matrices. Elements in A represent the similarities between sentences in the document set. All elements of i-th column in B are proportional to $rel(i|H)$. A and B are both normalized to make the sum of each row equal to 1. Note that k represents the kth iteration, and $p = [p_1, \ldots, p_N]^T$ is the vector of sentence ranking scores that we are looking for, which corresponds to the stationary distribution of the matrix M. The iteration is guaranteed to converge to a unique stationary distribution given that M is a stochastic matrix. To calculate the similarity of sentences in the document set, we use the cosine similarity. To calculate $rel(s, H)$, the sentence relevance given the human summary set, we use

$$rel(s, H) = \max_{r\in H} \text{ROUGE-2}_r(s), \tag{8}$$

where r is a sentence in the human summary, and ROUGE-$2_r(s)$ is the ROUGE-2 score of the sentence s with the reference r.

After estimating the score of every sentence in document set, a threshold is applied to assign a sentence rank 1 indicating summary sentence if the score is larger than the threshold, or otherwise rank 0 indicating non-summary sentence. The confidence score can be defined as

$$c_i = |p(x_i|H) - \text{threshold}|. \tag{9}$$

V. Feature Design

In our work, we use some common features that are widely used in the supervised summarization methods [9; 14] as well as several features induced from the unsupervised methods for learning the model. In total 20 features are used in our work.

[2]Since the ranks are estimated, more ranks may introduce more noise, and we show later on in our experiments that more ranks are not necessary.

A. Basic Features

The basic features are the commonly used features in previous summarization approaches, which can be extracted directly without complicated computation. Given a query and sentence pair, $< q, x_i >$, the basic features used for learning are described as follows.

Position: The position feature, denoted by Pos, indicates the position of x_i along the sentence sequence of a document. If x_i appears at the beginning of the document, Pos is set to be 1; if it is at the end of the document, Pos is 2; Otherwise, Pos is set to be 3.

Length: The length feature is the number of terms contained in x_i after removing the stop words according to a stop word list.

Number of Frequent Thematic Words: Thematic words are the most frequent words appeared in the documents after removing the stop words. Sentences containing more thematic words are more likely to be summary sentences. We use the number of frequent thematic words in x_i as a feature. In our work, 5 frequency thresholds 10,20,50,100, 200 are used to define the frequent thematic words, thus generating 5 features for each sentence.

Similarity to the Closest Neighboring Sentences: We also use the average similarity between a sentence and its closest neighbors as features. In particular, we use "Intra Sim to Pre N" and "Intra Sim to Next N" (N = 1, 2, 5) to record the average similarity of x_i to the previous N most similar sentences and to the next N most similar sentences respectively, in the same document. "Inter Sim to N" (N = 1,2,5) is also used to record the average similarity of x_i to the N most similar sentences in different documents. We use the cosine measure to compute the similarity measurement.

Similarity to the Query: The cosine similarity between the query q and the sentence x_i is also used as a feature.

B. Complex Features

Manifold Ranking Score: The ranking score is obtained for each sentence in the manifold-ranking process to denote the biased information richness of the sentence. All sentences in the document set plus the query description are considered as points $\{x_0, x_1, \ldots, x_n\}$ in a manifold space, where x_0 is the query description and the others are the sentences in the documents. The ranking function is denoted by $f = [f_0, f_1, \ldots, f_n]$. Since x_0 is the query description, the initial label vector of these sentences is $y = [y_0, y_1, \ldots, y_n]$, where $y_0 = 1, y_1 = \ldots = y_n = 0$. The manifold ranking can be computed iteratively using the following equation,

$$f(k+1) = \alpha S f(k) + (1-\alpha) y, \tag{10}$$

where S is the symmetrically normalized similarity matrix of $\{x_0, x_1, \ldots, x_n\}$, and α is a parameter, and k represents the k-th iteration. The iterative algorithm is guaranteed to converge to the final manifold ranking scores [7]. We set the α to 0.3, 0.5, 0.8 to obtain three different manifold ranking scores as three features. More detailed description of manifold ranking score can be found in [7].

C. Redundancy Removal

To generate the final summary, all our implemented methods use the diversity penalty algorithm as in [7] to impose redundancy penalty. as described in Algorithm 1. At each iteration of line 3-7, the sentence with the maximum score is selected into the summary, and other sentences are penalized according to their similarities to the selected sentence. A in line 7 indicates the normalized similarity matrix of all sentences.

Algorithm 1 Generate Final Summary

Input: sentence set: $S_1 = \{s_1, \ldots, s_n\}$,
scoring function: $f(s_i), 1 \leq i \leq n$,
Output: Summary: S_2

1: Initialize $S_2 = \emptyset$, score$(s_i) = f(s_i)$
2: **while** $S_1 \neq \emptyset$ and S_2 does not reach limit **do**
3: $s_{i^*} = \arg\max_{s \in S_1}$ score(s)
4: $S_1 = S_1 - \{s_{i^*}\}$
5: $S_2 = S_2 \cup \{s_{i^*}\}$
6: **for** s_j in S_1 **do**
7: score(s_j) = score$(s_j) - A_{ji^*} f(s_{i^*})$
8: **end for**
9: **end while**

VI. Experiments

A. Experiment Settings

We evaluate our proposed method for query-focused multi-document summarization on the main tasks of DUC 2005, DUC 2006, and DUC 2007. Each task has a gold standard data set consisting of document sets and reference summaries. In our experiments, DUC 2005 is used to train the model tested on DUC 2006, and DUC 2006 is used train the model tested on DUC 2007. Table II lists the characteristics of the data sets.

Table II
Brief description of the data sets.

	DUC 2005	DUC 2006	DUC 2007
#topics	50	50	45
#documents per topic	25-50	25	25
Summary length	250 words	250 words	250 words

We use ROUGE toolkit (version 1.5.5) [24] to measure the summarization performance. Several automatic evaluation methods are implemented in ROUGE, such as ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-SU. They measure the quality of a summary by counting the unit overlaps between the candidate summary and a set of reference summaries. ROUGE-N is an n-gram recall computed as follows:

Table III
SUMMARIZATION PERFORMANCE COMPARISON ON DUC 2006.

	ROUGE-1	ROUGE-2	ROUGE-W	ROUGE-SU
Ranking-SVM-CSL	0.4221 (0.4158-0.4279)	0.0994 (0.0949-0.1034)	0.1125 (0.1107-0.1143)	0.1542 (0.1503-0.1579)
Ranking-SVM	0.4215 (0.4155-0.4275)	0.0983 (0.0942-0.1026)	0.1128 (0.1111-0.1145)	0.1533 (0.1495-0.1560)
SVR	0.4166 (0.4104-0.4226)	0.0952 (0.0912-0.0992)	0.1106 (0.1088-0.1125)	0.1517 (0.1480-0.1555)
Manifold-Ranking	0.3882 (0.3821-0.3944)	0.0801 (0.0761-0.0842)	0.1043 (0.1028-0.1059)	0.1370 (0.1333-0.1409)
S24	0.4111 (0.4049-0.4171)	0.0951 (0.0909-0.0991)	0.1107 (0.1088-0.1125)	0.1547 (0.1506-0.1584)
S12	0.4048 (0.3992-0.4105)	0.0899 (0.0858-0.0939)	0.1079 (0.1061-0.1096)	0.1475 (0.1436-0.1514)
S23	0.4044 (0.3982-0.4097)	0.0879 (0.0837-0.0920)	0.1087 (0.1069-0.1103)	0.1449 (0.1410-0.1485)

Table IV
SUMMARIZATION PERFORMANCE COMPARISON ON DUC 2007.

	ROUGE-1	ROUGE-2	ROUGE-W	ROUGE-SU
Ranking-SVM-CSL	0.4496 (0.4435-0.4557)	0.1229 (0.1182-0.1270)	0.1177 (0.1158-0.1196)	0.1710 (0.1665-0.1758)
Ranking-SVM	0.4461 (0.4396-0.4526)	0.1203 (0.1158-0.1247)	0.1190 (0.1172-0.1207)	0.1701 (0.1658-0.1742)
SVR	0.4395 (0.4329-0.4466)	0.1179 (0.1132-0.1224)	0.1163 (0.1146-0.1182)	0.1652 (0.1607-0.1696)
Manifold-Ranking	0.3957 (0.3899-0.4022)	0.0769 (0.0733-0.0809)	0.1037 (0.1021-0.1055)	0.1362 (0.1329-0.1400)
S15	0.4451 (0.4379-0.4521)	0.1245 (0.1196-0.1293)	0.1194 (0.1174-0.1213)	0.1771 (0.1724-0.1818)
S29	0.4325 (0.4260-0.4387)	0.1203 (0.1155-0.1253)	0.1175 (0.1156-0.1194)	0.1707 (0.1609-0.1806)
S4	0.4342 (0.4291-0.4391)	0.1189 (0.1146-0.1237)	0.1154 (0.1137-0.1171)	0.1700 (0.1661-0.1754)

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{ref}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{ref}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}, \quad (11)$$

where n is the length of the n-gram, and ref stands for the reference summaries. $\text{Count}_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurring in a candidate summary and the reference summaries, and $\text{Count}(\text{gram}_n)$ is the number of n-grams in the reference summaries. ROUGE-L uses the longest common subsequence (LCS) statistics, while ROUGE-W is based on weighted LCS and ROUGE-SU is based on skip-bigram plus unigram. In the following experiments, we use ROUGE-1, ROUGE-2, ROUGE-W and ROUGE-SU, of which ROUGE-2 and ROUGE-SU were adopted by DUC 2006 and DUC 2007 for automatic performance evaluation, and all of which are widely used in summarization research.

SVM^{Rank} [25] is used as a tool for ranking SVM and also served as a basis for ranking SVM with cost sensitive loss. The parameter C in Eq.(1) and Eq.(4) is set to 1 for all following experiments, and other parameters are set to the default values. The threshold for assigning the two ranks to the sentences in training data generation is chosen by 10-fold cross validation.

B. System Comparison

First we compare our method Ranking-SVM-CSL (Ranking SVM with Cost Sensitive Loss) with three competitive baselines and three top systems of DUC. The baseline systems include 1) Ranking-SVM: applying ranking SVM directly; 2) SVR: learning a regression model using SVM; and 3) Manifold-Ranking: ranking the sentences according to the manifold ranking score, which is one of the features described in the previous section, where the parameter α is set to 0.5. All of the three baselines use the proposed graph based method in training data generation. The top three systems are the three systems with highest ROUGE-2 scores, chosen from the participant systems of DUC 2006 and DUC 2007, respectively, and are represented by their system IDs.

Table III and Table IV present the performance of these systems in ROUGE-1, ROUGE-2, ROUGE-W and ROUGE-SU along with corresponding 95% confidence intervals. As in [26], we approximately determine which differences in scores are significant via comparing the 95% confidence intervals, and significant differences are those where the confidence intervals for the estimates of the means for the two systems either do not overlap at all, or where the two intervals overlap but neither contains the best estimate for the mean of the other. From the results we can observe that our proposed method outperforms all baseline systems, performs significantly better than S12 and S23 on DUC 2006 and comparative to the two top systems S24 and S15 on DUC 2006 and DUC 2007 respectively, in most of ROUGE measures. It should be pointed out that the top systems in DUC involves much more preprocessing and postprocessing such as sentence reduction and entity deferencing in S15 [27].

Manifold-Ranking has the worst performance since it only uses the manifold ranking score as the single feature. Combination of multiple features leads to a significant improvement. Among the systems that automatically learn the combination weights for various features, learning to rank based methods (Ranking-SVM-CSL and Ranking-SVM) outperform the regression model (SVR). In particular, Ranking-SVM-CSL improves SVR significantly in respect of ROUGE-W on the DUC 2006 dataset and all except ROUGE-2 on the DUC 2007 dataset, while Ranking-SVM

improves SVR significantly only in respect of ROUGE-W on both datasets. Note that standard learning to rank methods focus on the ranking of the sentences and do not use the scores of the sentences. With Ranking-SVM-CSL, the scores of sentences are used as confidence in the loss function for sentence pairs, which leads to better performance than directly applying ranking SVM.

C. Training Data Generation Comparison

In this section, we empirically investigate the effects of different strategies for training data generation. We denote the proposed method of training data construction as graph-based-method and compare it with a set of baselines described below.

Given a summary set H for a query and a set of sentences $\{x_i\}_{i=1}^{N}$ in a set of documents, generally, the following strategy can be used to estimate the ranks of the sentences:

$$y_i^* = \max_{e \in H} y_{i,e}^* \tag{12}$$

where y_i^* is the estimated rank of sentence i, e is the reference which can be a sentence or a summary in H, $y_{i,e}^*$ is a discretized result of $sim(x_i, e)$ where sim can be the cosine similarity or ROUGE score of the sentence given the reference, representing the probability x_i is summary given the reference e.

We compare our graph-based method to this baseline strategy with different references (sentence or summary) and different similarity measurements (cosine similarity or ROUGE-2 score) and the comparison is shown in Figure 2. From the comparison, we observe that: 1) Using sentence as the reference is much better than using the whole summary, especially with the ROUGE score as the similarity function. This may due to the fact that more different words in the whole summary may lead to a bias in favor of those longer sentences having more overlapping grams with the reference, especially using similarity functions with no normalization factor, like ROUGE-2 score. 2) Our graph-based method outperforms other baseline strategies in most of combination of data and learning models. This is because our graph-based method makes use of the sentence relationships in the documents set, which has been shown as an important factor in a lot of summarization work to score the sentences.

D. Effect of Cost Sensitive Loss

In this section, we empirically investigate the effect of the cost sensitive loss. Figure 3(a) and Figure 3(b) show the performance comparison between Rank-SVM-CSL (with cost sensitive loss) and Ranking-SVM (without cost sensitive loss) for different thresholds on DUC 2006 and DUC 2007, respectively. For most thresholds we test, cost sensitive loss improves the performance on both DUC 2006 and DUC 2007. We can observe that the performance of Ranking SVM, especially in Figure 3(b) changes frequently with the variation of the threshold. Compared with directly using ranking SVM, the results of Ranking-SVM-CSL are more stable.

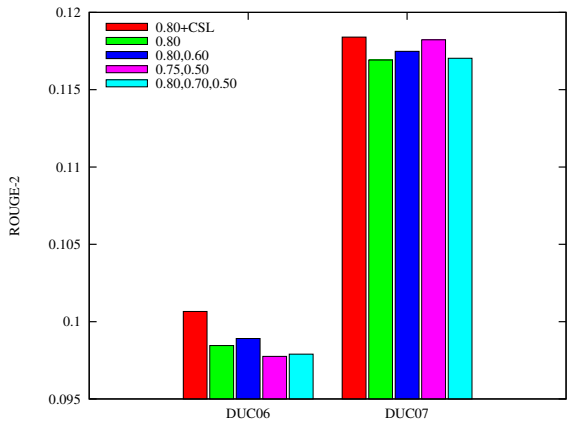


Figure 4. Performance comparison using training data with multiple ranks.

E. Granularity of Rank

In our work, the sentences of the document set are divided into two ranks: summary and non-summary. Here we use a case study to show that more ranks do not lead to significant performance improvements. Instead of using only one threshold (0.8 in this case), we map the sentences to more than two ranks by selecting more than one thresholds. Intuitively, the number of summary sentences should be less than the number of non-summary sentences. Hence the thresholds are chosen to make the number of sentences in a higher rank less than that in a lower rank.

Figure 4 shows the performance using ranking SVM using different thresholds. "+CSL" indicates learning with ranking SVM with cost sensitive loss. We observe that: although using 3 or more ranks (i.e., with 2 or more thresholds) may lead to better results (e.g., (0.80,0.60) on DUC 2006 and DUC 2007, (0.75,0.50) on DUC 2007, and (0.80,0.70,0.50) on DUC 2007), the improvement is unstable and small, compared with the improvement made by 0.80+CSL (i.e., using threshold 0.8 followed by learning with ranking SVM with cost sensitive loss). We leave it as future work to explore the effects of applying cost sensitive loss to cases with more than two ranks.

VII. Conclusion and Future Work

In this paper, we explore the use of a learning to rank approach, ranking SVM to combine features for extractive query-focus multi-document summarization. To apply ranking SVM for summarization, we propose a graph-based method for training data generation by utilizing the sentence relationships and introduce a cost sensitive loss to improve the robustness of learning. The experiments demonstrate the effectiveness of our proposed methods. In our future work, we will use more complex features in feature design, and explore other learning to rank algorithms.

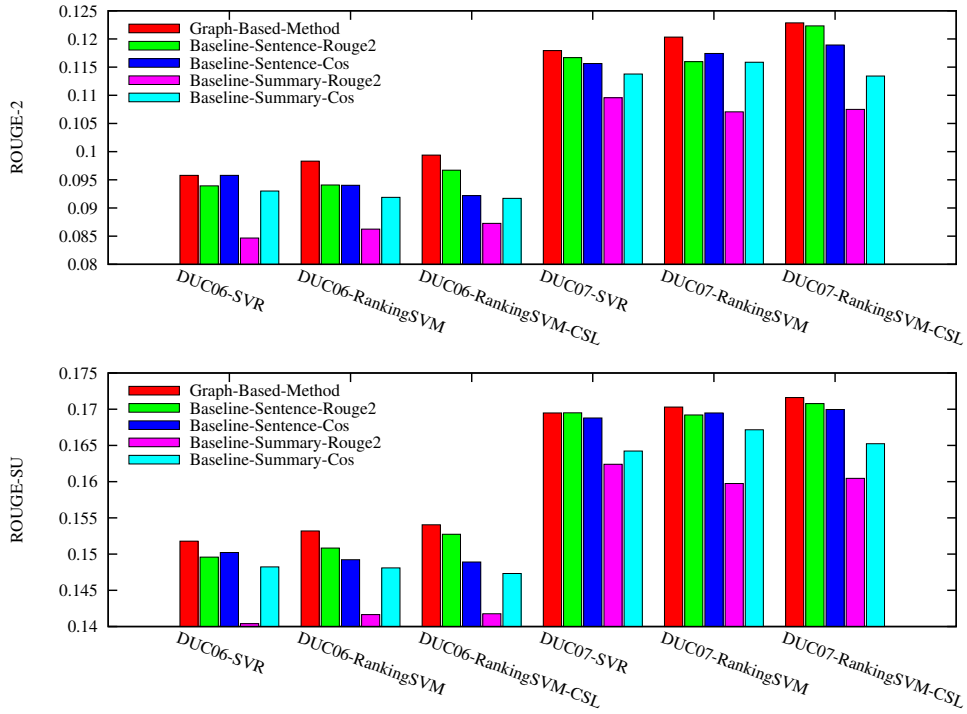
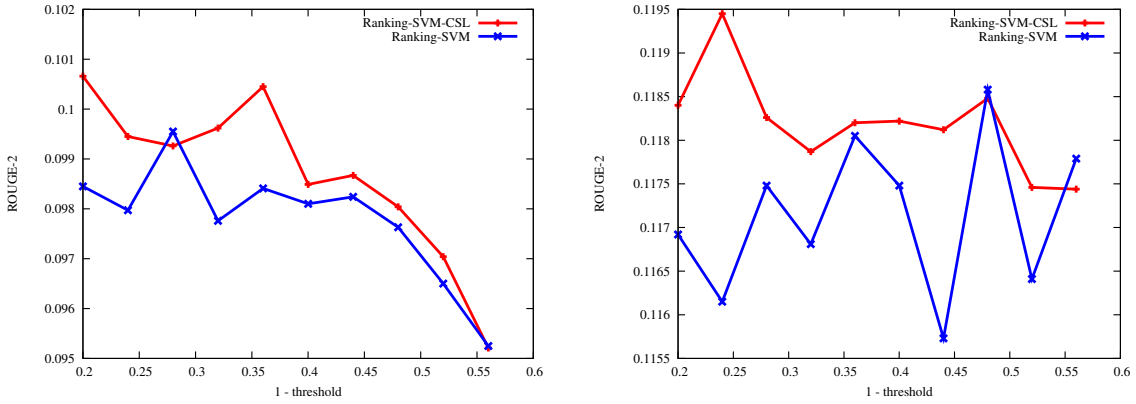


Figure 2. Performance comparison of training data generation.



(a) DUC 2006 (b) DUC 2007

Figure 3. Effects using cost sensitive loss. (Value of x-axis represents $1 - \text{threshold}$)

ACKNOWLEDGMENT

The work is partially supported by NSF grants DMS-0915110, CCF-0830659, and HRD-0833093.

REFERENCES

[1] 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.

[2] V. Nastase, "Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 763–772.

[3] C. Long, M. Huang, X. Zhu, and M. Li, "Multi-document summarization by information distance," in *Proceedings of the 9th IEEE International Conference on Data Mining*. IEEE, 2009, pp. 866–871.

[4] D. Wang, T. Li, S. Zhu, and C. Ding, "Multi-document summarization via sentence-level semantic analysis and sym-

metric matrix factorization," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 307–314.

[5] C. Shen and T. Li, "Multi-document summarization via the minimum dominating set," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 984–992.

[6] J. Otterbacher, G. Erkan, and D. Radev, "Using random walks for question-focused sentence retrieval," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2005, pp. 915–922.

[7] X. Wan, J. Yang, and J. Xiao, "Manifold-ranking based topic-focused multi-document summarization," in *Proceedings of the 20th international joint conference on Artifical intelligence*. Morgan Kaufmann Publishers Inc., 2007, pp. 2903–2908.

[8] D. Jurafsky and J. Martin, *Speech and language processing*. Prentice Hall New York, 2008.

[9] Y. Ouyang, S. Li, and W. Li, "Developing learning strategies for topic-based summarization," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 79–86.

[10] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 133–142.

[11] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1995, pp. 68–73.

[12] T. Hirao, H. Isozaki, E. Maeda, and Y. Matsumoto, "Extracting important sentences with support vector machines," in *Proceedings of the 19th international conference on Computational linguistics*. Association for Computational Linguistics, 2002, pp. 1–7.

[13] L. Zhou and E. Hovy, "A web-trained extraction summarization system," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Association for Computational Linguistics, 2003, pp. 205–211.

[14] D. Shen, J. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *Proceedings of the 20th international joint conference on Artifical intelligence*, vol. 7, 2007, pp. 2862–2867.

[15] L. Zhao, X. Huang, and L. Wu, "Fudan University at DUC 2005," in *Proceedings of DUC*, vol. 2005, 2005.

[16] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 129–136.

[17] M. Taylor, J. Guiver, S. Robertson, and T. Minka, "Softrank: optimizing non-smooth rank metrics," in *Proceedings of the international conference on Web search and web data mining*. ACM, 2008, pp. 77–86.

[18] T. Liu, "Learning to rank for information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[19] Y. Freund, R. Iyer, R. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *The Journal of Machine Learning Research*, vol. 4, pp. 933–969, 2003.

[20] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 89–96.

[21] J. Xu, Y. Cao, H. Li, and Y. Huang, "Cost-sensitive learning of SVM for ranking," *Machine Learning: European Conference on Machine Learning*, pp. 833–840, 2006.

[22] Y. Cao, J. Xu, T. Liu, H. Li, Y. Huang, and H. Hon, "Adapting ranking SVM to document retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 186–193.

[23] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," *Advances in Neural Information Processing Systems*, pp. 115–132, 1999.

[24] C. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Association for Computational Linguistics, 2003, pp. 71–78.

[25] T. Joachims, "Svm-rank: Support vector machine for ranking," *http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html*, 2009.

[26] A. Nenkova, L. Vanderwende, and K. McKeown, "A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 573–580.

[27] P. Pingali, K. Rahul, and V. Varma, "IIIT Hyderabad at DUC 2007," in *Proceedings of DUC 2007*, 2007.