A Comparative Study on Ranking and Selection Strategies for Multi-Document Summarization

Feng Jin, Minlie Huang, Xiaoyan Zhu
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Dept. of Computer Science and Technology, Tsinghua University
jinfengfeng@gmail.com,{aihuang,zxy-dcs}@tsinghua.edu.cn

Abstract

This paper presents a comparative study on two key problems existing in extractive summarization: the ranking problem and the selection problem. To this end, we presented a systematic study of comparing different learning-to-rank algorithms and comparing different selection strategies. This is the first work of providing systematic analysis on these problems. Experimental results on two benchmark datasets demonstrate three findings: (1) pairwise and listwise learning-to-rank algorithms outperform the baselines significantly; (2) there is no significant difference among the learning-to-rank algorithms; and (3) the integer linear programming selection strategy generally outperformed Maximum Marginal Relevance and Diversity Penalty strategies.

1 Introduction

As the rapid development of the Internet, document summarization has become an important task since document collections are growing larger and larger. Document summarization, which aims at producing a condensed version of the original document(s), helps users to acquire information that is both important and relevant to their information need. So far, researchers have mainly focused on extractive methods which choose a set of salient textual units to form a summary. Such textual units are typically sentences, sub-sentences (Gillick and Favre, 2009), or excerpts (Sauper and Barzilay, 2009).

Almost all extractive summarization methods face two key problems: the first problem is how to rank textual units, and the second one is how to select a subset of those ranked units. The ranking problem requires systems model the relevance of a textual unit to a topic or a query. In this paper, the ranking problem refers to either sentence ranking or concept ranking. Concepts can be unigrams, bigrams, semantic content units, etc., although in our experiment, only bigrams are used as concepts. The selection problem requires systems improve diversity or remove redundancy so that more relevant information can be covered by the summary as its length is limited. As our paper focuses on extractive summarization, the selection problem refers to selecting sentences. However, the selection framework presented here is universal for selecting arbitrary textual units, as discussed in Section 4.

There have been a variety of studies to approach the ranking problem. These include both unsupervised sentence ranking (Luhn, 1958; Radev and Jing, 2004, Erkan and Radev, 2004), and supervised methods (Ouyang et al., 2007; Shen et al., 2007; Li et al., 2009). Even given a list of ranked sentences, it is not trivial to select a subset of sentences to form a good summary which includes diverse information within a length limit. Three common selection strategies have been studied to address this problem: Maximum Marginal Relevance (MMR) (Carbonell and Goldstein, 1998), Diversity Penalty (DiP) (Wan, 2007), and integer linear programming (ILP) (McDonald, 2007; Gillick and Favre, 2009). As different methods were often evaluated on different datasets, it is of great value to systematically compare ranking and selection strategies on the same dataset. However, to the best of our knowledge, there is still no work to compare different ranking strategies or compare different selection strategies.

In this paper, we presented a comparative study on the ranking problem and the selection

Coling 2008: Poster Volume, pages 525–533,
Beijing, August 2010

problem for extractive summarization. We compared three genres of learning-to-rank methods for ranking sentences or concepts: SVR, a pointwise ranking algorithm; RankNet, a pairwise learning-to-rank algorithm; and ListNet, a listwise learning-to-rank algorithm. We adopted an ILP framework that is able to select sentences based on sentence ranking or concept ranking. We compared it with other selection strategies such as MMR and Diversity Penalty. We conducted our comparative experiments on the TAC 2008 and TAC 2009 datasets, respectively. Our contributions are two-fold: First, to the best of our knowledge, this is the first work of presenting systematic and in-depth analysis on comparing ranking strategies and comparing selection strategies. Second, this is the first work using pairwise and liswise learning-to-rank algorithms to perform concept (word bigram) ranking for extractive summarization.

The rest of this paper is organized as follows. We introduce the related work in Section 2. In Section 3, we present three ranking algorithms, SVR, RankNet, and ListNet. We describe the sentence selection problem with an ILP framework described in Section 4. We introduce features in Section 5. Evaluation and experiments are presented in Section 6. Finally, we conclude this paper in Section 7.

2 Related Work

A number of extractive summarization studies used unsupervised methods with surface features, linguistic features, and statistical features to guide sentence ranking (Edmundson, 1969; McKeown and Radev, 1995; Radev et al., 2004; Nekova et al., 2006). Recently, graph-based ranking methods have been proposed for sentence ranking and scoring, such as LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004).

There are also a variety of studies on supervised learning methods for sentence ranking and selection. Kupiec et al. (1995) developed a naive Bayes classifier to decide whether a sentence is worthy to extract. Recently, Conditional Random Field (CRF) and Structural SVM have been employed for single document summarization (Shen et al., 2007; Li et al., 2009).

Besides ranking sentences directly, there are some approaches that select sentences based on concept ranking. Radev et al. (2004) used centroid words whose *tf***idf* scores are above a threshold. Filatova and Hatzivassiloglou (2004) used atomic event as concept. Moreover, summarization evaluation metrics such as Basic Element (Hovy et al., 2006), ROUGE (Lin and Hovy, 2003) and Pyramid (Passonneau et al., 2005) are all counting the concept overlap between generated summaries and human-written summaries.

Another important issue existing in extractive summarization is to find an optimal sentence subset which can cover diverse information. Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) and Diversity Penalty (Wan, 2007) are most widely used approaches to reduce redundancy. The two methods are essentially based on greedy search. By contrast, ILP based approaches view summary generation as a global optimization problem. McDonald (2007) proposed a sentence-level ILP solution. Sauper and Barzilay (2009) presented an excerpt-level ILP method to generate Wikipedia articles. Gillick and Favre (2009) proposed a concept-level ILP, but they used document frequency to score concepts (bigrams), without any learning process. Some recent studies (Gillick and Favre, 2009; Martins and Smith, 2009) also modeled sentence selection and compression jointly using ILP. Our ILP framework proposed here is based on these studies. Although various selection strategies have been proposed, there is no work to systematically compare these strategies yet.

Learning to rank attracts much attention in the information retrieval community recently. Pointwise, pairwise and listwise learning-to-rank approaches have been extensively studied (Liu, 2009). Some of those have been applied to document summarization, such as SVR (Ouyang et al., 2007), classification SVM (Wang et al., 2007), and RankNet (Svore et al., 2007). Again, there is no work to systematically compare these ranking algorithms. To the best of our knowledge, this is the first time that a listwise learning-to-rank algorithm, ListNet (Cao et al., 2007), is adapted to document summarization in this paper. Moreover, pairwise and listwise learning-to-rank algorithms have never been used to perform concept ranking for extractive summarization.

3 Ranking Sentences or Concepts

Given a query and a collection of relevant documents, an extractive summarization system is required to generate a summary consisting of a set of text units (usually sentences). The first problem we need to consider is to determine the importance of these sentences according to the input query. We approach this ranking problem in two ways: the first way is to score sentences directly using learning-to-rank algorithms, and thus the goal of summarization is to select a subset of sentences, considering both relevance and redundancy. The second way is to score concepts within the document collection, and then the summarization task is to select a sentence subset that can cover those important concepts maximally. The problem of sentence selection will be described in Section 4.

Suppose the relevant document collection for a query q is D_q. From this collection, we obtain a set of sentences or concepts (e.g., word bigrams), $S=\{s_1,s_2,...,s_n\}$ or $C=\{c_1,c_2,..., c_n\}$. Before training, each s_i or c_i is associated with a gold standard score, y_i. A feature vector, $x_j=\Phi(s_j/c_j,q,D_q)$, is constructed for each sentence or concept. The learning algorithm will learn a ranking function $f(x_j)$ from a collection of query-document pairs $\{(q_i,D_{qi})|i= 1, 2,...,m\}$.

We investigated three learning-to-rank methods to learn $f(x_j)$. The first one is a pointwise ranking algorithm, support vector regression (SVR). This algorithm treats sentences (or concepts) independently. The second method is a pairwise ranking algorithm, RankNet, which learns a ranking function from a list of sentence (or concept) pairs. Each pair is labeled as 1 if the first sentence s_i (or concept c_i) ranks ahead of the second s_j (or c_j), and 0 otherwise.

The listwise ranking algorithm, ListNet, learns the ranking function $f(x_j)$ in a different way. A list of sentences (or concepts) is treated as a whole. Both RankNet and ListNet take into account the dependency between sentences (or concepts).

3.1 Support Vector Regression

Support Vector Regression (SVR), a generalization of the classical SVM formulation, attempts to learn a regression model. SVR has been applied to summarization in (Ouyang et al., 2007; Metzler and Kanungo, 2008). In our work, we train the SVR model to fit the gold standard score of each sentence or concept.

Formally, the objective of SVR is to minimize the following objective:

$$\Im(w,b,\xi) = \left\{\frac{1}{2}\| w\|^2 + C\left(v\cdot\xi + \frac{1}{N}\sum_{x_i} L(y_i - f(x_i))\right)\right\} \quad (1)$$

where $L(x)=|x|-\xi$ if $x > \xi$ and otherwise $L(x)=0$; y_i is the gold standard score of x_i; $f(x) =w^Tx+b$, the predicted score of x; C and v are two parameters; and N is the total number of training examples.

3.2 RankNet

RankNet is a pairwise learning-to-rank method (Burges et al., 2005). In this algorithm, training examples are handled pair by pair. Given a pair of feature vectors (x_i, x_j), the gold standard probability $\overline{P}_{ij}$ is set to be 1 if the label of the pair is 1, which means x_i ranks ahead of x_j. The gold standard probability is 0 if the label of the pair is 0. Then the predicted probability P_{ij}, which defines the probability of x_i ranking ahead of x_j by the model, is represented as a logistic function:

$$P_{ij} = \frac{\exp(f(x_i) - f(x_j))}{1+\exp(f(x_i) - f(x_j))} \quad (2)$$

where $f(x)$ is the ranking function. The objective of the algorithm is to minimize the cross entropy between the gold standard probability and the predicted probability, which is defined as follows:

$$C_{ij}(f) = -\overline{P}_{ij}\log P_{ij} - (1-\overline{P}_{ij})\log(1-P_{ij}) \quad (3)$$

A three-layer neural network is used as the ranking function, as follows:

$$f(x_n) = g^3\left(\sum_j w_{ij}^{32} g^2\left(\sum_k w_{jk}^{21} x_{nk} + b_j^2\right) + b_i^3\right) \quad (4)$$

where for weights w and bias b, the superscripts indicate the node layer while the subscripts indicate the node indexes within each layer. And x_{nk} is the k-th component of input feature vector x_n. Then a gradient descent method is used to learn the parameters. For details, refer to (Burges et al., 2005).

3.3 ListNet

ListNet takes a list of items as input in the learning process. More specifically, suppose we have

a list of feature vectors ($x_1, x_2, ..., x_n$) and each feature vector x_i has an gold standard score y_i, which has been assigned before training. Accordingly, we have a list of gold standard scores ($y_1, y_2, ..., y_n$). We also have a list of scores assigned by the algorithm during training, say, ($f(x_1), f(x_2), ..., f(x_n)$). Given a score list $S=\{s_1, s_2, ..., s_n\}$, the probability that x_j will rank the first place among the n items is defined as follows:

$$P_s(j) = \frac{\Phi(s_j)}{\sum_{k=1}^{n}\Phi(s_k)} = \frac{\exp(s_j)}{\sum_{k=1}^{n}\exp(s_k)} \quad (5)$$

It is easy to prove that ($P_s(1), P_s(2), ..., P_s(n)$) is a probability distribution, as the sum of them equals to 1. Therefore, the cross entropy can be used to define the loss between the gold standard distribution $P_y(j)$ and the distribution $P_f(j)$, as follows:

$$L(y, f) = -\sum_{j=1}^{n} P_y(j)\log P_f(j) \quad (6)$$

where y represents the gold standard score list ($y_1, y_2, ..., y_n$) and $f=(f(x_1), f(x_2), ..., f(x_n))$ is the score list output by the ranking algorithm.

The function f is defined as a linear function, as follows:

$$f_w(x_i) = w^T x_i \quad (7)$$

Then the gradient of loss function $L(y,f)$ with respect to the parameter vector w can be calculated as follows:

$$\Delta w = \frac{\partial L(y, f_w)}{\partial w} = -\sum_{j=1}^{n} P_y(x_j)\frac{\partial f_w(x_j)}{\partial w} + \frac{1}{\sum_{j=1}^{n}\exp(f_w(x_j))}\sum_{j=1}^{n}\exp(f_w(x_j))\frac{\partial f_w(x_j)}{\partial w} \quad (8)$$

During training, w is updated in a gradient descent manner: $w=w-\eta\Delta w$ and η is the learning rate. For details, refer to (Cao et al., 2007).

4 ILP-based Selection Framework

After we have a way of ranking sentences or concepts, we face a sentence selection problem: selecting an optimal subset of sentences as the final summary. To integrate sentence/concept ranking, we adopted an integer linear programming (ILP) framework to find the optimal sentence subset (Filatova and Hatzivassiloglou, 2004; McDonald, 2007; Gillick and Favre, 2009; Takamura and Okumura, 2009). ILP is a global optimization problem whose objective and constraints are linear in a set of integer variables.

Formally, we define the problem of sentence selection as follows:

$$\text{maximize: } \left\{\sum_i f(x_i) * z_i^x\right\} \quad (9)$$

$$s.t. \quad \sum_j z_j^u * |u_j| \le Lim$$

$$\sum_j z_j^u * I(i,j) \ge z_i^x, \quad \forall i$$

$$(z_i^x + z_j^x) * sim(x_i, x_j) < \delta \quad \forall i, j$$

$$z_i^x, z_j^u \in \{0,1\}, \quad \forall i, j$$

where:

$\boldsymbol{x_i}$ – the representation unit, such as a sentence or a concept. We term it representation unit because the summary quality is represented by the set of included x_i;

$\boldsymbol{f(x_i)}$ - the ranking function given by the learning-to-rank algorithms;

$\boldsymbol{u_j}$ - the selection unit, for instance, a sentence in this paper. $|u_j|$ is the number of words in u_j;

$\boldsymbol{z_i^x}$ - the indicator variable which denotes the presence or absence of x_i in the summary;

$\boldsymbol{z_j^u}$ - the indicator variable which denotes inclusion or exclusion of u_j;

$\boldsymbol{I(i, j)}$ - a binary constant indicating that whether x_i appears in u_j. It is either 1 or 0;

Lim - the length limit;

$\boldsymbol{sim(x_i, x_j)}$ - a similarity measure for considering the redundancy;

$\boldsymbol{\delta}$ - the redundancy threshold.

The first constraint indicates the length limit. The second constraint asserts that if a representation unit x_i is included in a summary, at least one selection unit that contains x_i must be selected. The third constraint considers redundancy. If the representation unit is sentence, the similarity measure is defined as *tf*idf* similarity, and $\delta/2$ is the similarity threshold, which was set to be 1 here. For concepts, the similarity measure can be defined as

$$sim(x_i, x_j) = \begin{cases} 1, & x_i = x_j \\ 0, & \text{otherwise} \end{cases}$$

However, other definition is also feasible, depending on what has been selected as representation unit.

Note that this framework is very general. If the representation unit x_i is a sentence, the ranking function is defined on sentence. Thus the ILP framework will find a set of sentences that can optimize the total scores of selected sentences, subject to several constraints. If the representation unit is a concept, the ranking function measures the importance of a concept to be included in a summary. Thus the goal of ILP is to find a set of sentences by maximizing the scores of concepts covered by those selected sentences.

D_q	relevant document collection in response to query q
d	one single document
w_i	unigram
w_iw_{i+1}	bigram
S	sentence
$tf_d(w_i)$	the frequency of w_i occurring in d
$df_D(w_i)$	the number of documents containing w_i in collection D

Table 1. Notations for features.

5 Features

To facilitate the following description, some notations are defined in Table 1. In our dataset, each query has a title and narrative to precisely define an information need. The following is a query example from the TAC 2008 test dataset:

```
<topic id = "D0801A">
  <title> Airbus A380 </title>
  <narrative>
      Describe developments in the production and
      launch of the Airbus A380.
  </narrative>
</topic>
```

Features for sentence ranking and concept ranking are listed in the following. We use word bigrams as concept here.

Sentence Features

(1) Cluster frequency: $\sum_{w_i \in S} tf_{D_q}(w_i)$

(2) Title frequency: $\sum_{w_i \in S} tf_d(w_i)$ where d is a new document that consists of all the titles of documents in D_q.

(3) Query frequency: $\sum_{w_i \in S} tf_d(w_i)$ where d is a document consisting of the title and narrative fields of the current topic.

(4) Theme frequency: $\sum_{w_i \in S \wedge w_i \in T} tf_{D_q}(w_i)$ where T is the top 10% frequent unigram words in D_q.

(5) Document frequency of bigrams in the sentence: $\sum_{w_iw_{i+1} \in S} df_D(w_iw_{i+1})$.

(6) PageRank score: as described in (Mihalcea and Tarau, 2004), each sentence in D_q is a node in the graph and the cosine similarity between a pair of sentences is used as edge weight.

Concept Features

(1) Cluster frequency: $tf_{D_q}(w_iw_{i+1})$, the frequency of w_iw_{i+1} occurring in D_q.

(2) Title frequency: $tf_d(w_iw_{i+1})$, where d is a document consisting of all the titles of documents in D_q.

(3) Query Frequency: the frequency of the bigram occurring in the topic title and narrative.

(4) Average term frequency: $\sum_{d \in D_q} tf_d(w_iw_{i+1}) / |D_q|$. $|D_q|$ is the number of documents in the set.

(5) Document frequency: the document frequency of this bigram.

(6) Minimal position: the minimal position of this bigram relative to the document length.

(7) Average position: the average position of this bigram in collection D_q.

6 Experimental Results

6.1 Data Preprocessing

We conducted experiments on the TAC 2008 and TAC 2009 datasets. The task requires producing a 100-word summary for each query (also termed topic sometimes). There are 48 queries in TAC 2008 and 44 queries in TAC 2009. A query example has been given in Section 5. Relevant documents for these queries have been specified. And four human-written summaries were supplied as reference summaries for each query.

We segmented the relevant documents into sentences using the LingPipe toolkit [1] and stemmed words using the Porter Stemmer. Word bigrams are used as concepts in this paper. If the two words in a bigram are both stopwords, the bigram will be discarded. The sen-

[1] http://alias-i.com/lingpipe/index.html

tence features and bigram features are then calculated. As our focus is on comparing different ranking strategies and selection strategies, we did not apply any sophisticated linguistic or semantic processing techniques (as pre- or post-processing). Thus we did not compare our results to those submitted to the TAC conferences.

We train the learning algorithms on one dataset and then evaluate the algorithms on the other. The generated summaries are evaluated using the ROUGE toolkit (Lin and Hovy, 2003).

6.2 Preparing Training Samples

As our work includes both sentence ranking and concept ranking, we need to establish two types of training data. Fortunately, we are able to do this based on the reference summaries and annotation results provided by the TAC conferences.

For the sentence ranking problem, we compute the average ROUGE-1 score for each sentence by comparing it to the four reference summaries for each query. This score is treated as the gold-standard score. In ListNet, these scores are directly used (see formula (5)). While in RankNet, the sentences for a query are grouped into 10 bins according to their ROUGE-1 scores, and then we extract sentences from different bins respectively to form a pair. We assume that a sentence in a higher scored bin should rank ahead of those sentences in lower scored bins.

As for the concept ranking problem, gold-standard scores are obtained from the human annotated Pyramid data. The weight of each semantic content unit (SCU) is the number of reference summaries in which the SCU appears. So straightforwardly, the gold-standard score of a bigram is the largest weight of all SCUs that contain the bigram. And if a bigram does not occur in any SCU, its score will be 0. Thus the bigram scores belong to the set {0,1,2,3,4} as there are four human-written summaries for each query. These scores are directly used in ListNet (see formula (5)). And in RankNet, bigram pairs are constructed according to the gold-standard scores.

6.3 Learning Parameters

For SVR, the radial basis kernel function is employed and the optimal values for parameters *C*, *v* and *g* (for the kernel) are found using the *gridregression.py* tool provided by LibSVM (Chang and Lin, 2001) with a 5-fold cross validation on the training set.

RankNet applies a three-layer (one hidden layer) neural network with only one node in the output layer, as described in (Burges et al., 2005). The number of hidden neurons was empirically set to be 10. The learning rate was set to 0.001 for sentence ranking and 0.01 for bigram ranking.

As for ListNet, the learning rate for sentence ranking and concept ranking are both set to be 0.1 empirically.

6.4 Comparing Ranking Strategies

In this section, we compared different ranking strategies for both sentence ranking and concept ranking. The sentence selection strategies were fixed to the ILP selection framework as shown in Section 4. We chose ILP as the selection strategy because we want to compare our system with the following two methods (as baselines):

(1) ***SENT_ILP***: A sentence-level method proposed by McDonald (2007) with ILP formulation. We implemented the query-focused version of the formulae as TAC 2008 and 2009 required query-focused summarization.

(2) ***DF_ILP***: A concept-level ILP method using document frequency to score word bigrams (Gillick and Favre, 2009), without any learning process.

The differences between our framework and SENT_ILP are: a) SENT_ILP used a redundancy factor in the objective function whereas we modeled redundancy as constraints; b) SENT_ILP used *tf***idf* similarity to compute relevance scores whereas we used learning algorithms.

The ROUGE-1 and ROUGE-2 measures for each method are presented in Table 2 and Table 3. Note that the performance on the TAC 2008 dataset was obtained from the models that were trained on the TAC 2009 dataset. Then, the datasets were interchanged for training and testing, respectively. Different learning-to-rank strategies (SVR, RankNet, ListNet) do not show significant differences between one and another, but they all outperform *SENT_ILP* substantially (p-value < 0.0001). And for concept ranking, RankNet and ListNet both achieve significantly better ROUGE-2 results (p-value < 0.005) than

DF_ILP. This infers that considering more features will have better results than using document frequency to score concepts. The Wilcoxon signed-rank test (Wilcoxon, 1945) is used for significance tests in our experiment. A good ranking strategy for modeling relevance is important for extractive summarization. RankNet which used a three-layer network (non-linear function) as the ranking function performs slightly better than ListNet which is based on a linear ranking function.

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	SVR	0.35086	0.08447
	RankNet	**0.36025**	**0.09291**
	ListNet	0.35365	0.09129
	SENT_ILP	0.31546	0.06500
TAC 2009	SVR	0.36125	0.09659
	RankNet	**0.36216**	**0.09778**
	ListNet	0.35480	0.09126
	SENT_ILP	0.31962	0.07034

Table 2. Results of sentence ranking strategies.

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	SVR	0.36555	0.10291
	RankNet	**0.37564**	**0.11213**
	ListNet	0.36863	0.10660
	DF_ILP	0.36922	0.10373
TAC 2009	SVR	0.37126	0.10698
	RankNet	**0.37513**	**0.11364**
	ListNet	0.37499	0.11313
	DF_ILP	0.36347	0.10156

Table 3. Results of concept ranking strategies.

It is worth noting that Pyramid annotations may not cover all important bigrams, partly because SCUs in reference summaries have been rephrased by human annotators. Note that we simply extract original sentences to form a summary, thus it is possible that a bigram which is important in the original sentences does not appear in any rephrased SCUs at all. Such bigrams will have a gold-standard score of 0, which is erroneous supervision. For example, the bigrams *hurricane katrina* in topic D0804A about *Katrina pet rescue* and *life support* in D0806A about *Terri Schiavo case* are not annotated in any SCUs, but these bigrams are both key terms for the topics.

6.5 Comparing Selection Strategies

In order to study the influence of different selection strategies, we compare the ILP selection strategy (as introduced in Section 4) with other popular selection strategies, based on the same sentence ranking algorithm (we chose sentence-level RankNet). The baselines to be compared are as follows:

(1) ***MMR***: As shown in (Carbonell and Goldstein, 1998), the formula of MMR is:

$$MMR = \arg\max_{s_i \in R-S} \left\{ \lambda D_1(q, s_i) - (1-\lambda) \max_{s_j \in S} D_2(s_i, s_j) \right\}$$

where q is the given query; R is the set of all sentences; S is the set of already included sentences; D_1 is the normalized ranking score $f(x_i)$ of s_i, and D_2 is the cosine similarity of the feature vectors for s_i and s_j. Our implementation was similar to the MMR strategy in the MEAD[2]summarizer.

(2) ***DiP***: Diversity penalty which penalizes the score of candidate sentences according to the already selected ones (Wan, 2007).

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	ILP	**0.36025**	**0.09291**
	MMR	0.35459	0.09086
	DiP	0.35263	0.08689
TAC 2009	ILP	**0.36216**	**0.09778**
	MMR	0.35148	0.08881
	DiP	0.34714	0.08672

Table 4. Comparing selection strategies.

The corresponding ROUGE scores are presented in Table 4. ILP outperforms other selection strategies significantly on the TAC 2009 dataset (both ILP vs. MMR and ILP vs. DiP). Although improvements are observed with ILP on the TAC 2008 dataset, the difference is not significant (using ILP vs. using MMR). MMR is comparable to DiP as they are both based on greedy search in nature.

To investigate the difference between these strategies, we present in-depth analysis here. First, the average length of summaries generated by ILP is 97.1, while that by MMR and DiP are 95.5 and 92.7, respectively. Note that the required summary length is 100 and that more words can potentially cover more information. Thus, ILP can generate summaries with more information. This is because ILP is a global optimization algorithm, subject to the length constraint. Second, the average rank of sentences selected by ILP is 12.6, while that by MMR and

[2] http://www.summarization.com/mead/

DiP is about 5, which is substantially different. ILP can search down the ranked list while the other two methods tend to only select the very top sentences. Third, there are 4.1 sentences on average in each ILP-generated summary, while the number for MMR and DiP generated summaries are 2.7 and 2.5, respectively. Thus ILP tend to select shorter sentences than MMR and DiP. This may help reduce redundancy as longer sentences may contain more topic irrelevant clauses or phrases.

6.6 Discussions

Interestingly, although the learning-to-rank algorithms combined with the ILP selection strategy perform well in summarization, the performance is still far from that of manual summarization. In this study, we investigate the upper bound performance. We used the presented ILP framework to generate summaries based on the gold-standard scores, rather than the scores given by the learning algorithms. In other words, $f(x_i)$ in formula (9) is replaced by the gold-standard scores. The ROUGE results are shown in Table 5. We also listed the best/worst/average ROUGE scores of human summaries in TAC by comparing one human summary (as generated summary) to the other three human summaries (as reference summaries). These results are substantially better than those by the learning algorithms. Sentence- and concept- level ranking produces very close results to best human summaries. Some ROUGE-2 scores are even higher than those of human summaries. This is reasonable as human annotators may have difficulty in organizing content when there are many documents and sentences. The results reflect that there is a remarkable gap between the gold-standard scores and the learned scores.

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	Sentence-level	0.44216	0.14842
	Concept-level	0.42222	**0.16018**
	Human Best	**0.44220**	0.13079
	Human Average	0.41417	0.11606
	Human Worst	0.38005	0.10736
TAC 2009	Sentence-level	0.45500	0.15565
	Concept-level	0.43526	**0.17118**
	Human Best	**0.45663**	0.14864
	Human Average	0.44443	0.12680
	Human Worst	0.39652	0.11109

Table 5. Upper bound performance.

7 Conclusion and Future Work

We presented systematic and extensive analysis on studying two key problems in extractive summarization: the ranking problem and the selection problem. We compared three genres of learning-to-rank algorithms for the ranking problem, and investigated ILP, MMR, and Diversity Penalty strategies for the selection problem. To the best of our knowledge, this is the first work of presenting systematic comparison and analysis on studying these problems. We also at the first time proposed to use learning-to-rank algorithms to perform concept ranking for extractive summarization.

Our future work will focus on: (1) exploiting more features that can reflect summary quality; (2) optimizing summarization evaluation metrics directly with new learning algorithms.

Acknowledgments

This work was partly supported by the Chinese Natural Science Foundation under grant No. 60973104 and No. 60803075, and with the aid of a grant from the International Development Research Center, Ottawa, Canada IRCI project from the International Development.

References

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton and Greg Hullender. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai and Hang Li. 2007. Learning to Rank: from Pairwise Approach to Listwise Approach. In *Proceedings of ICML 2007*.

Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of SIGIR*, August 1998, pp. 335 - 336.

Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: a Library for Support Vector Machines. Software available at *http://www.csie.ntu.edu.tw/~cjlin/libsvm*

H. P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the ACM (JACM) Archive*, Volume 16, Issue 2 (April 1969) Pages: 264 - 285.

G. Erkan and Dragomir R. Radev. 2004. LexPageRank: Prestige in Multi-Document Text Summa-

rization. In *Proceedings of EMNLP 2004*, Barcelona, Spain.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based Extractive Summarization. In *Proceedings of ACL Workshop on Summarization*, volume 111.

Dan Gillick and Benoit Favre. 2009. A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*.

Eduard Hovy, Chin-yew Lin, Liang Zhou and Junichi Fukumoto. 2006. Automated Summarization Evaluation with Basic Elements. In *Proceedings of the Fifth Conference on Language Resources and Evaluation*.

Julian Kupiec, Jan Pedersen and Francine Chen. 1995. A Trainable Document Summarizer. In *Proceedings of SIGIR'95*, pages 68 - 73, New York, USA.

Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha and Yong Yu. 2009. Enhancing Diversity, Coverage and Balance for Summarization through Structure Learning. In *Proceedings of the 18th International Conference on World Wide Web*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In *Proceedings of HLT-NAACL*, pages 71-78.

Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval, Foundation and Trends on Information Retrieval. Now Publishers.

H.P. Luhn. 1958. The Automatic Creation of Literature Abstracts. In *IBM Journal of Research and Development*, Vol. 2, No. 2, pp. 159-165, April 1958.

André F. T. Martins and Noah A. Smith. 2009. Summarization with a Joint Model for Sentence Extraction and Compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*\.

Ryan McDonald. 2007. A Study of Global Inference Algorithms in Multi-Document Summarization. In *Proceedings of the 29th ECIR*.

Kathleen McKeown and Dragomir R. Radev. 1995. Generating Summaries of Multiple News Articles. In *Proceedings of SIGIR'95*, pages 74–82.

Donald Metzler and Tapas Kanungo. 2008. Machine Learned Sentence Selection Strategies for Query-Biased Summarization. *SIGIR Learning to Rank Workshop*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP 2004*, Barcelona, Spain, July 2004.

Ani Nenkova, Lucy Vanderwende and Kathleen McKeown. 2006. A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors that Influence Summarization. In *Proceedings of SIGIR 2006*.

You Ouyang, Sujian Li, Wenjie Li. 2007. Developing Learning Strategies for Topic-based Summarization. In *Proceedings of the sixteenth ACM Conference on Information and Knowledge Management*, 2007.

Rebecca J. Passonneau, Ani Nenkova, Kathleen McKeown and Sergey Sigelman. 2005. Applying the Pyramid Method in DUC 2005. *DUC 2005 Workshop*.

Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. 2004. Centroid-based Summarization of Multiple Documents. *Information Processing and Management*, 40:919–938.

Christina Sauper and Regina Barzilay. 2009. Automatically Generating Wikipedia Articles: A Structure-Aware Approach. In *Proceedings of ACL* 2009.

Dou Shen, Jian-Tao Sun, Hua Li, QiangYang and Zheng Chen. 2007. Document Summarization Using Conditional Random Fields. In *IJCAI*, pages 2862 - 2867, 2007.

Krysta Svore, Lucy Vanderwende, and Chris Burges. 2007. Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources. In *Proceedings of EMNLP-CoNLL (2007)*, pp. 448-457..

Hiroya Takamura and Manabu Okumura. Text Summarization Model Based on Maximum Coverage Problem and its Variant. In *Proceedings EACL*, 2009.

Xiaojun Wan and Jianguo Xiao. 2007. Towards a Unified Approach Based on Affinity Graph to Various Multi-document Summarizations. *ECDL 2007*, 297-308.

Changhu Wang, Feng Jing, Lei Zhang and Hong-Jiang Zhang. 2007. Learning Query-Biased Web Page Summarization. In *Proceedings of the sixteenth ACM Conference on Information and Knowledge Management*.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. Biometrics, 1, 80-83.