iRANK: A Rank-Learn-Combine Framework for Unsupervised Ensemble Ranking

Furu Wei
IBM Research–China, Beijing and Department of Computing, The Hong Kong Polytechnic University, Hong Kong, P.R. China. E-mail: weifuru@cn.ibm.com

Wenjie Li
Department of Computing, The Hong Kong Polytechnic University, Hong Kong, P.R. China. E-mail: cswjli@comp.polyu.edu.hk

Shixia Liu
IBM Research–China, Beijing, P.R. China. E-mail: liusx@cn.ibm.com

The authors address the problem of unsupervised ensemble ranking. Traditional approaches either combine multiple ranking criteria into a unified representation to obtain an overall ranking score or to utilize certain rank fusion or aggregation techniques to combine the ranking results. Beyond the aforementioned "combine-then-rank" and "rank-then-combine" approaches, the authors propose a novel "rank-learn-combine" ranking framework, called Interactive Ranking (iRANK), which allows two base rankers to "teach" each other before combination during the ranking process by providing their own ranking results as feedback to the others to boost the ranking performance. This mutual ranking refinement process continues until the two base rankers cannot learn from each other any more. The overall performance is improved by the enhancement of the base rankers through the mutual learning mechanism. The authors further design two ranking refinement strategies to efficiently and effectively use the feedback based on reasonable assumptions and rational analysis. Although iRANK is applicable to many applications, as a case study, they apply this framework to the sentence ranking problem in query-focused summarization and evaluate its effectiveness on the DUC 2005 and 2006 data sets. The results are encouraging with consistent and promising improvements.

Introduction

Ranking plays an important role in information retrieval and natural language processing applications. Many factors (a.k.a. features) have been taken into account when designing the ranking functions (or the rankers), which results in the demand for a mechanism to integrate the features. There are two alternative integration approaches in the literature. One is to first combine the features into a unified representation, and then use it to rank the text segments. The other is to utilize the rank fusion or rank aggregation techniques to combine the ranking results (scores, ranks, or orders) produced by the multiple ranking functions into a unified rank. The second approach is also known as ensemble ranking, the most popular implementation of which is to linearly combine the ranking features to obtain an overall score, which is then used as the ranking criterion. The weights of the features are either experimentally tuned or automatically derived by applying certain learning-based mechanisms. However, both of the above-mentioned "combine-then-rank" and "rank-then-combine" approaches have a common drawback. They do not make full use of the information provided by the different ranking functions and neglect the interactions among them before combination. We believe that each individual ranking function (we call it a *base ranker*) is able to provide valuable information to the other base rankers such that they can learn from each other by means of mutual ranking refinement, which, in turn, may result in an overall improvement in ranking. To the best of our knowledge, this is a research area that has not been well addressed in the past.

The inspiration for the work presented in this article comes from the idea of cotraining (Blum & Mitchell, 1998), which is a very successful paradigm in the semisupervised learning framework for classification. In essence, cotraining employs two weak classifiers that help augment each other to boost the performance of learning algorithms. Two classifiers mutually

Received October 8, 2009; accepted December 7, 2009

 • Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.21296

cooperate with each other by providing their own labeling results to enrich the training data for the other parties during the supervised learning process. The interactions between the two classifiers is the use of classifier-labeled data to incrementally create pseudotraining data, which may not be 100% correct but sufficient to provide more information for training a better classifier without requiring human participation. Unlike classification, labeled data is difficult to obtain for ranking problems in most cases. However, we impart the spirit of cotraining in the context of ranking. Although each base ranker cannot decide the overall ranking well by itself, its ranking results indeed reflect its opinion towards the ranking from its point of view. The two base rankers can then share their own opinions by providing the ranking results to each other as feedback. For each ranker, feedback from other rankers contains additional information to guide the refinement of its own ranking results if feedback is defined and used appropriately. This process continues iteratively until the two base rankers cannot learn from each other any more. We call this kind of ranking paradigm interactive ranking (iRANK). The way to use the feedback information varies depending on the nature of a particular ranking task.

iRANK is applicable to many applications. In this article, we are particularly interested in the task of query-focused summarization, in which sentence ranking is the issue of most concern upon the extractive summarization framework. Up to now, the feature-based ranking approaches, which rank sentences based on the features elaborately designed to characterize sentences, have been among the most effective and popular approaches. As different features may reflect different aspects of the sentences, we therefore expect a framework to combine them together to produce significant overall ranking results. For this purpose, we design a new sentence ranking algorithm in which a query-dependent ranker and a query-independent ranker mutually learn from each other upon the iRANK framework.

The main contributions of this work are threefold.

1. We present a "rank-learn-combine" unsupervised ensemble ranking framework, namely, interactive ranking (iRANK).
2. We explore two ranking refinement strategies that utilize the feedback either as an additional ranking feature or to ensure rank consistency during refinement.
3. We propose two new sentence-ranking algorithms based on iRANK for query-focused summarization and evaluate their effectiveness on the DUC 2005 and 2006 data sets.

The remainder of this article is organized as follows. In the next section, we briefly review related work on rank fusion and aggregation and related work on sentence ranking in summarization. We then introduce the proposed interactive ranking framework and present an application of this framework in query-focused summarization. We report experiments and evaluation results on the DUC data sets and conclude with some open questions.

Related Work

Rank Fusion and Aggregation

Rank fusion or aggregation (a.k.a. ensemble ranking) is the problem of combining a set of ranking lists in such a way to optimize the performance of the combination. Existing approaches in general fall into three categories, namely unsupervised, semisupervised, and supervised approaches.

Unsupervised rank aggregation has been extensively investigated in the context of meta-search, where the most critical problem was to combine the ranked lists of document returns by multiple search engines in response to a given query. This problem can be naturally decomposed into three subproblems: (a) normalizing relevance scores given by the input stems, (b) estimating the relevance scores of unretrieved documents, and (c) combining the newly acquired scores for each document into one improved score. Aslam and Montague (2001) investigated metasearch models based on an optimal democratic voting procedure (called the *Borda Count*), Bayesian inference, and a model for obtaining upper bounds on the performance of metasearch algorithms. In other work (Montague & Aslam, 2001), they showed that the techniques used for normalizing relevance scores and estimating the relevance scores of unretrieved documents could have a significant effect on the overall performance of a metasearch.

As for the application of metasearch, Dwork, Kumar, Naor, and Sivakumar (2001) addressed the problem of combining ranking results from various sources in the context of the Web. They developed a set of rank aggregation techniques and compared their performance to that of well-known methods. Montague and Aslam (2002) proposed a new algorithm, called *Condorcet-fuse*, to improve retrieval results by combining document ranking functions. Beginning with one of the two major classes of voting procedures from social choice theory (i.e. the Condorcet procedure), they applied a graph-theoretic analysis that yielded an elegant, efficient, and effective sorting based algorithm. Aslam, Pavlu, and Savell (2007) presented a unified model, which, given the ranked lists of documents returned by multiple retrieval systems, simultaneously solved the problems of (a) fusing the ranked lists of documents to obtain a high-quality combined list, (b) generating document collections likely to contain large fractions of relevant documents; and (c) accurately evaluating the underlying retrieval systems with small numbers of relevance judgments. Farah and Vanderpooten (2007) focused on the task where rankings of documents were searched in the same collection and were provided by multiple methods. They proposed a multiple criteria framework using an aggregation mechanism based on the decision rules that identified positive/negative reasons for judging whether a document should get a better rank than another. In database applications, such as catalog searches and fielded searches, etc., Fagin, Kumar, and Mahdian (2004) provided a comprehensive picture of how to compare partial rankings. They proposed several metrics to compare partial rankings, presented algorithms that efficiently compute

DOI: 10.1002/asi

them, and proved that they are within constant multiples of each other.

Besides unsupervised rank aggregation approaches, semisupervised, and supervised approaches were also studied recently in the context of information retrieval. In the camp of semisupervised rank aggregation research, many ensemble-ranking approaches were proposed to learn appropriate weights for combing multiple rankers. For example, Hoi and Jin (2008) learned query-dependent weights to combine multiple rankers in document retrieval to overcome the shortcoming in existing methods, i.e., the learned weights were query independent. Chen, Wang, Song, and Zhang (2008) learned a ranking function based on the ordering agreement of different rankers. To improve the accuracy of rank aggregation, Liu, Liu, Qin, Ma, and Li (2007) employed a supervised learning approach, in which an order-based aggregation function was trained within an optimization framework using the labeled data. Despite the effectiveness of the supervised learning approaches to rank aggregation, Klementiev, Roth, and Small (2008) argued that learning required the supervised ranked data, which was expensive to acquire.

As a matter of fact, in the community of traditional information retrieval, typical formalisms such as the vector space model, the best-match model, and the language model tended to first combine features (such as term frequency and document length) into a unified representation, and then used the unified representation to rank documents. Pickens and Colovchinsky (2008) took the opposite approach. Documents were first ranked by the relevance of a single feature value and were assigned scores based on their relative ordering within the collection. A separate ranked list was created for every feature value; these lists were then fused to produce a final document scoring. This new "rank-then-combine" approach was extensively evaluated and was shown to be as effective as traditional "combine-then-rank" approaches. This observation drives us to investigate and design more effective mechanisms for integrating the rankers or the features.

Finally, we note that the work presented in this work is somewhat related to a recent work by Jin, Valizadegan, and Li (2008), in which they addressed the problem of ranking refinement, namely, improving the accuracy of an existing ranking function with a small set of labeled instances. Although only one ranker was considered in their work, multiple (two) rankers are involved in this study. Our work substantially differs in that we employ the iterative mutual learning among different base rankers in an unsupervised manner. The idea of unsupervised ranking refinement process can be found as a part of our framework, but the strategies are totally different because the motivations and assumptions are distinctly different.

Sentence Ranking in Summarization

Feature-based sentence-ranking approaches are widely used in document summarization. They have been extensively investigated in the past due to their easy implementation and the ability to achieve promising results. The use of feature-based ranking has led to many successful (e.g., top five) systems in DUC (Document Understanding Conference) 2005–2007 query-focused summarization evaluations (Over, Dang, & Harman, 2007). A variety of statistical and linguistic features, such as sentence length, sentence position, named entity, etc., can be found in literature. Among them, query term feature, centroid (Radev, Jing, Stys, & Tam, 2004), and signature term (Lin & Hovy, 2000) are most remarkable. Schilder and Kondadadi (2008) proposed a query-focused summarization system based solely on word-frequency features of clusters, documents, and topics, which achieved comparable accuracy to the best systems presented in recent DUC evaluations. The features were often linearly combined and the weights of them were either experimentally tuned (Li, Li, Li, Chen, & Wu, 2005) or automatically derived by applying a certain learning-based mechanism (Ouyang, Li, & Li, 2007; Wong, Wu, & Li, 2008). Learning-based approaches, such as the discriminative training model, the support vector regression (SVR) model, and the log-linear model, etc., were popular in recent DUC competitions (DUC Reports), and have achieved encouraging results in DUC 2007.

Graph-based sentence-ranking algorithms have also drawn much attention in document summarization. LexRank (Erkan & Radev, 2004) for generic summarization and query-sensitive LexRank for query-focused summarization (Otterbacher, Erkan, & Radev, 2005) modeled a document or a set of documents as a weighted text graph by taking the sentences from the document(s) as nodes and the similarity between two sentences as edge weight. Different from feature-based approaches, graph-based approaches took into account the global information and recursively calculated sentence significance from the entire text graph rather than only relying on unconnected individual sentences. The effectiveness of these approaches came from the advantage of making use of the link structure information. Many publications on extending existing LexRank-like algorithms can be found in the literature. Here we only give references to the most recent ones (Wan & Yang, 2008; Wei, Li, Lu, & He, 2008).

Interactive Ranking Framework

Motivation and Problem Statement

The traditional approach to integrate multiple ranking results from different individual rankers is to combine the ranking results (e.g., scores or ranks) produced by the individual rankers through certain rank aggregation techniques. A problem with existing rank aggregation approaches (ensemble ranking) is their assumption that the rankers do not communicate with each other, and as a result they lose the opportunity to revise (or to refine) their own ranking results before combination. Let $X = \{x_1, x_2, \cdots, x_n\}$ be the instances to be ranked. f_i is the i-th base ranker, $f_i(x) \rightarrow \Re$, $\forall x \in X$. Our goal is then to jointly improve multiple base rankers through the interactions among them, which can

DOI: 10.1002/asi

consequently result in the overall improvement in ranking reliability and accuracy.

Interactive Ranking (iRANK) Framework

To this end, we develop an interactive ranking (iRANK) framework. For the sake of explanation, let's consider two base rankers here. Given a set of instances X, one can define the two base ranker f_1 and f_2 for some intended purposes. The ranking results produced by f_1 and f_2 individually are by no means perfect. However, either f_1 or f_2 can provide relatively reasonable ranking information to "teach" each other so as to jointly improve them. In such a collaborative teach-and-learn mode, the ranking becomes an interactive process. One way to do the interactive ranking is to take the most confident ranking results (e.g. highly ranked instances based on scores, ranks, or orders) from one base ranker as the feedback to update the other's ranking results, and vice versa. This process continues iteratively until the termination condition is reached. It is noteworthy that the standard cotraining algorithm requires two sufficient and redundant views, i.e., the attributes be naturally portioned into two sets, each of which is sufficient for learning and conditionally independent to the other given the class label. In view of this, it is suggested that f_1 and f_2 are defined as two independent rankers which emphasize different aspects of the instances X. The framework of interactive ranking is depicted in Procedure 1 below.

Procedure 1. iRANK(f_1, f_2, X, ϕ)

1. Rank X with f_1 and obtain the ranking results r_1^*.
2. Rank X with f_2 and obtain the ranking results r_2^*.
3. Normalize r_1^*, $r_1^*(x_i) \leftarrow \dfrac{r_1^*(x_i) - \min(r_1^*)}{\max(r_1^*) - \min(r_1^*)}$.
4. Normalize r_2^*, $r_2^*(x_i) \leftarrow \dfrac{r_2^*(x_i) - \min(r_2^*)}{\max(r_2^*) - \min(r_2^*)}$.
5. $r_1 \leftarrow r_1^*(x_i)$, $r_2 \leftarrow r_2^*(x_i)$.
6. Repeat.
7. Choose the top N ranked instances τ_1^n at round n from r_1 as feedback to supervise f_2, and re-rank X using f_2 and τ_1; Update r_2;
   $r_2 \leftarrow \phi(f_2, \tau_1)$, $\tau_1 = \tau_1^{(1)} \cup \tau_1^{(2)} \cup \cdots \cup \tau_1^{(n)}$
8. Choose the top N ranked instances τ_2^n at round n from r_2 as feedback to supervise f_1, and re-rank X using f_1 and τ_2; Update r_1;
   $r_1 \leftarrow \phi(f_1, \tau_2)$, $\tau_2 = \tau_2^{(1)} \cup \tau_2^{(2)} \cup \cdots \cup \tau_2^{(n)}$
9. Until I(X).
10. $r(x_i) = \lambda \cdot r_1(x_i) + (1 - \lambda) \cdot r_2(x_i)$.
11. Return r.

Notice that the interactive ranking process can be clearly divided into three main steps.

1. Rank: Run the two base rankers f_1 and f_2 and obtain the initial ranking lists r_1 and r_2 (Steps 1–2).
2. Learn: The two base rankers refine their ranking results by sharing feedback with each other (Steps 7–8). The learning process here is both interactive and iterative.
3. Combine: When the two base rankers cannot learn from each other any more, return the combination results directly (Step 10). Finally, we can use the traditional linear combination strategy with λ as the combination factor. Note that in this work the initial ranking results are normalized before the learning process, which therefore ensures the final ranking results (scores) comparable among different rankers.

What distinguishes iRANK from the traditional rank-then-combine approaches is that it involves a learning process in an unsupervised manner before the combination. Although the feedback τ can be defined in different ways depending on the nature of the application, such as the ranking scores, the ranks, the instances, or the combination of the aforementioned information, we consider the top-ranked instances as the feedback in this study. In addition, the ranking refinement ϕ can be defined variously in different context and the termination condition I(X) can be defined according to the different application scenarios. We address the ranking refinement strategies in the next section and the termination conditions in the following section.

Ranking Refinement Strategy

In particular, we investigate two ranking refinement strategies, which utilize the feedback in different ways.

Feedback as a feature (iRANK-FF). A simple way is to use the feedback as an additional feature in ranking refinement. In the analysis that follows, we hold the following assumption:

The instances that are similar to the highly ranked instances should also be highly ranked.

This can be formulated as,

$$\begin{cases} r_1(x) \leftarrow \eta \cdot r_1^*(x) + (1 - \eta) \cdot \pi_1(x) \\ r_2(x) \leftarrow \eta \cdot r_2^*(x) + (1 - \eta) \cdot \pi_2(x) \end{cases} \quad (1)$$

where η is a balance factor that can be viewed as the proportion of dependence of the new ranking results on its initial ranking. Equation 1 indicates that the refined ranking of the instance x from one base ranker (say f_1) consists of two parts. The first part is the initial ranking (i.e., $r_1^*(x)$) produced by f_1. The second part is the similarity between x and the top N feedback instances provided by the other ranker (say f_2). Because the top N-ranked instances by f_2 are supposed to be highly supported by f_2, an instance that is similar to them should also deserve a high rank. Through this mutual interaction, the two base rankers "teach" each other and are expected as a whole to produce more reliable ranking results.

π_1 and π_2 in Equation 1 are the scores generated from the feedback τ_2 and τ_1, respectively, which captures the

DOI: 10.1002/asi

effect of the feedback and the definition of them may vary. For example, they can be defined as the maximum, the minimum or the average similarity value between x and the feedback instances in τ_2 or τ_1. As mentioned before, learning is iterative. From the n rounds of iterations, one can actually collect a series of n τ_1 and τ_2. We certainly can use only the feedback obtained at the current round. However, we believe that it is more robust and reliable to use the whole collection of feedback from the historical interaction records especially when the number of instances in the feedback information is very small. In this paper, we define τ_1 and τ_2 as,

$$\begin{cases} \pi_1(x) \leftarrow \dfrac{\sum_{k=1}^{n} sim(x, \tau_2^{(k)})}{n}, \ \pi_1 = \dfrac{\pi_1 - \min(\pi_1)}{\max(\pi_1) - \min(\pi_1)} \\ \pi_2(x) \leftarrow \dfrac{\sum_{k=1}^{n} sim(x, \tau_1^{(k)})}{n}, \ \pi_2 = \dfrac{\pi_2 - \min(\pi_2)}{\max(\pi_2) - \min(\pi_2)} \end{cases} \tag{2}$$

where $sim(x, \tau_2^{(k)})$ or $sim(x, \tau_1^{(k)})$ denotes the similarity between the instance x and the feedback $\tau_2^{(k)}$ or $\tau_1^{(k)}$ at the round k. It can be defined as the maximum, the minimum or the average similarity value between x and the feedback instances in $\tau_2^{(k)}$ or $\tau_1^{(k)}$. We assume that each base ranker is most confident with its first ranked sentence and set N to 1 in iRANK-FF. As a result, $sim(x, \tau_1^{(k)})$ or $sim(x, \tau_2^{(k)})$ is defined as the similarity between x and the only one sentence in $\tau_2^{(k)}$ or $\tau_1^{(k)}$.

Feedback for consistent rank learning (iRANK-CRL). Recall that our ultimate goal is to generate the unified (thus consistent) ranking results from the two base rankers. An alternative way to use the feedback is to learn a consistent rank from both the initial ranking results and the feedback in the process of ranking refinement. The development of this consistency driven ranking refinement strategy is motivated by the following observation:

Similar instances have similar ranks, and the refined ranking results should subject, as consistently as possible, to its own initial ranking results and the rank of the similar instances in the feedback it receives.

We model the above-mentioned intuition as follows. At the round $k+1$, for the two base rankers, we formally formulate two cost functions $\Psi_1(r)$ and $\Psi_2(r)$ in a joint regularization framework similar to the one proposed in Zhou, Bousquet, Lal, Weston, and Scholkopf (2003) as follows,

$$\Psi_1(r) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij}\left\|\frac{1}{\sqrt{\Delta_{ii}}} r_1^{(k+1)}(i) - \frac{1}{\sqrt{\Delta_{jj}}} r_2^{(k)}(j)\right\|^2 + \mu\sum_{i=1}^{n}\left\|r_1^{(k+1)}(i) - r_1^{(*)}(i)\right\|^2\right) \tag{3}$$

$$\Psi_2(r) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij}\left\|\frac{1}{\sqrt{\Delta_{ii}}} r_2^{(k+1)}(i) - \frac{1}{\sqrt{\Delta_{jj}}} r_1^{(k)}(j)\right\|^2 + \mu\sum_{i=1}^{n}\left\|r_2^{(k+1)}(i) - r_2^{(*)}(i)\right\|^2\right) \tag{4}$$

where $r_1^{(k)}$ and $r_2^{(k)}$ denote the ranking scores of the instances in τ_1 and τ_2 at the round k, respectively. The optimized ranking refinement strategy is the one that minimizes the regularization functions given in Equations 3 and 4.

Let us take the objective function in Equation 3 as an example. The first term on its right-hand side is the "smoothness constraint" on the ranks in the feedback π_2, which means that the refined ranking should be consistent to the corresponding ranks of the similar instances in π_2. The second term is the "fitting constraint" to its initial ranking, which means that the refined ranking should not change too much from the initial ranking. μ is a positive number used as the tradeoff between the two competing constraints. Differentiating $\Psi_1(r)$ with respect to r, we have

$$\left.\frac{\partial \Psi_1}{\partial r}\right|_{r=r^*} = r^* - \mathrm{H}\cdot r_2 + u(r^* - r_1^{(*)}) = 0 \tag{5}$$

where $\mathrm{H} = \Delta^{-1/2} W \Delta^{-1/2}$, W is the similarity matrix among the sentences and Δ is a diagonal matrix with $\Delta_{ii} = \sum_j W_{ij}$. We have

$$r^* = \frac{1}{1+u}\mathrm{H}\cdot r_2^{(k)} + \frac{u}{1+u} r_1^{(*)} \tag{6}$$

Let $\alpha = \frac{1}{1+u}$, $\beta = \frac{u}{1+u}$, then we have,

$$r^* = \alpha \cdot \mathrm{H} \cdot r_2^{(k)} + \beta \cdot r_1^{(*)} \tag{7}$$

Note that $\alpha + \beta = 1$. Finally, we have

$$r^* = (1-\beta)\cdot \mathrm{H} \cdot r_2^{(k)} + \beta \cdot r_1^{(*)} \tag{8}$$

$$r_1^{(k+1)} = \phi(f_1, \tau_2) = (1-\beta)\cdot \mathrm{H} \cdot r_2^{(k)} + \beta \cdot r_1^{(*)} \tag{9}$$

$r_2^{(k+1)}$ is obtained in a similar way. As seen from Equation 9, the refined ranking results consist of two parts. The second term $\beta \cdot r_1^{(*)}$ reflects a straightforward consistency with the initial ranking results of f_1, while the first term $(1-\beta)\cdot \mathrm{H}\cdot r_2^{(k)}$ comes from the ranking results of the highly ranked instances in τ_2 from f_2. The first term can be interpreted in this way. In the ranking process of f_1, an instance which is relevant to a high rank instance evaluated by f_2 earns a bonus score that steps up its initial rank. This can be viewed as the process of f_1 learning consistent ranking from f_2. Compared to the ranking refinement strategy proposed in the preceding section, the consistency driven ranking refinement strategy differs in two aspects. First, the relative importance

DOI: 10.1002/asi

of the instances in the feedback is taken into account. The instances in the feedback are treated equally in iRANK-FF, whereas the instances in the feedback are discriminated by their ranking scores in iRANK-CRL. Second, only the top one ranked instance is concerned in iRANK-FF; however, the top N instances are used as the feedback[1] in iRANK-CRL. We believe that the results produced by this new strategy should be more reasonable and reliable because Equation 9 not only well captures the given intuition, more important, it is also mathematically described by an optimization problem presented in Equation 3. This conclusion is validated through the experiments given in the Experiment and Evaluation section below.

Termination Analysis

We now address the termination condition $\mathrm{I}(X)$ in the iRANK framework. We set up the termination strategy as follows.

The iterative mutual learning process should be terminated *iff* (1) the two base ranker can not learn from each other any more, or (2) there is no useful feedback available for learning.

Regardless of the ranking refinement strategies, either iRANK-FF or iRANK-CRL, iRANK terminates when the top K instances in r_1 and r_2 are identical. This is because we are particularly interested in the top ranked instances. It is also very likely that r_1 and r_2 do not change any more after several rounds of iterations. In this case, the two base rankers cannot learn from each other any more and iRANK should be terminated.

As in iRANK-FF, we always consider the first ranked instance as the feedback. However, if the instances to be considered have been used already in previous iterations, we will move to the second best instances. We do not continue this searching process because using the instances in lower rank as feedback means to introduce noise, which, in turn, may result in even worse ranking results after mutual learning. This strategy also ensures that iRANK-FF will terminate when there is no appropriate feedback available for learning. As for iRANK-CRL, we can easily prove its termination. Take f_1 for example, we have $r_1^{(i+1)} - r_1^{(i)} = (1-\beta)^{i-j} \cdot \mathrm{H}^{i-j} \cdot (r_1^{(j+1)} - r_1^{(j)}), \quad \forall i \geq j \geq 0,$ thus, $\lim_{i\to\infty} |r_1^{(i+1)} - r_1^{(i)}| = \lim_{i\to\infty} |(1-\beta)^{i+1} \cdot \mathrm{H}^i \cdot (\mathrm{H} \cdot r_2^{(*)} - r_1^{(*)})| = 0$.[2] Please see the Appendix for the detailed proof of the convergence and termination of iRANK-CRL. As a result, r_1 does not change any more after several rounds of iterations, which indicates the two base rankers cannot learn from each other any more, and thus should be terminated. It is the same for r_2. In summary, iRANK is guaranteed to terminate through the aforementioned termination conditions.

Application of iRANK in Query-Focused Summarization

Task Definition of DUC Query-Focused Summarization

The query-oriented multidocument summarization task defined in the DUC evaluations requires generating a concise and well-organized summary for a set of the relevant documents according to a given query that simulates a user's information need. The query usually consists of one or more interrogative and/or narrative sentences. Here is a query example from the DUC 2005 document set "d331f."

<topic>
<num> d331f </num>
<title> World Bank criticism and response </title>
<narrative>
Who has criticized the World Bank and what criticisms have they made of World Bank policies, activities or personnel. What has the Bank done to respond to the criticisms?
</narrative>
<granularity> specific </granularity>
</topic>

According to the task definitions, system-generated summaries are strictly limited to 250 words in length.

In this article, we follow the traditional sentence extraction-based summarization framework, where the most critical processes involved are sentence ranking and sentence selection. We present the sentence ranking algorithm in the next section and the sentence selection strategy in the section thereafter.

Sentence Ranking Based on iRANK

To design the sentence-ranking algorithm based on the proposed iRANK framework, it is most important to design the two base rankers and to define the details of ranking refinement.

In the context of query-focused summarization, two kinds of features, i.e., query-dependent and query-independent features, are necessary and they are supposed to complement each other. We then use these two kinds of features to develop the two base rankers. The query-dependent feature (i.e., the relevance of the sentence s to the query q) is defined as the cosine similarity between s and q,

$$f_1 \Leftrightarrow rel(s,q) = \cos(s,q) = \overrightarrow{s} \bullet \overrightarrow{q} / \| \overrightarrow{s} \| \cdot \| \overrightarrow{q} \| \quad (10)$$

Here the words in the sentences and query are weighted by *tf-isf*, where *tf* is the word frequency in the sentence or query, and $isf_w = \log(N_S^D / sf_w)$ is the inverse sentence frequency (ISF) of w, where sf_w denotes the sentence frequency

[1] We use the feedback from the last round because the number of the instances in the feedback information here is big enough and it reflects and implements the intuition of iRANK-CRL.

[2] Because $0 < \beta < 1$, and the eigenvalues of H are in $[-1, 1]$, therefore $\lim_{i\to\infty} ((1-\beta) \cdot \mathrm{H})^i = 0$. As for the convergence threshold, we set it as 0.000001 in the experiments.

DOI: 10.1002/asi

of w, and N_S^D denotes the total number of sentences in the document set D.

As for the query-independent feature, we consider the LexRank as the base ranker. Let $LR(s_i)$ denotes the LexRank score of the sentence s_i, the ranker can be iteratively computed as follow.

$$f_2 \Leftrightarrow LR(s_i) = (1-d)\cdot 1/n + d\cdot \sum_{s_i \in D \cap i \neq j} LR(s_j)\cdot sim(s_i, s_j) \quad (11)$$

where d is the damping factor and $sim(s_i, s_j)$ is the cosine similarity between the two sentences s_i and s_j. We rewrite Equation 11 in the matrix style as

$$f_2 \Leftrightarrow M\cdot LR = \lambda \cdot LR \quad M = d\cdot W + (1-d)\cdot \frac{1}{n}\cdot e\cdot e^T \quad (12)$$

where LR denotes a ranking vector and W is the sentence affinity matrix. Accordingly, LR (i.e., f_2) can be computed as the corresponding eigenvector of the maximum eigenvalue of M (i.e., 1). We can use the power iteration method to acquire LR.

Note that the query-relevant ranker is a link-unaware base ranker. Only the sentence itself is concerned in the ranking process. The LexRank ranker, on the other hand, is a link-aware base ranker, in which the links or to say the relationships among the sentences are taken into consideration.

For the first ranking refinement strategy (i.e., iRANK-FF), we only need to specify the similarity measure,

$$sim(s, \tau) = sim(s, s^{(*)}) = \vec{s} \bullet \overrightarrow{s^{(*)}} / \|\vec{s}\| \cdot \|\overrightarrow{s^{(*)}}\| \quad (13)$$

where $s^{(*)}$ denotes the sentence in the feedback τ. The corresponding sentence ranking algorithm is illustrated in Algorithm 1.

Algorithm 1. iRANK-FF(f_1, f_2, D, q)

1. Extract sentences $S = \{s_1, \ldots s_m\}$ from D.
2. Define the ranking refinement strategy,

$$\begin{cases} r_1(x) \leftarrow \eta \cdot r_1^*(x) + (1-\eta)\cdot \pi_1(x) \\ r_2(x) \leftarrow \eta \cdot r_2^*(x) + (1-\eta)\cdot \pi_2(x) \end{cases} \quad (14)$$

   where τ_1 and τ_2 denote the similarity scores to the sentences in the feedback as specified in Equation 2.
3. Return iRANK(f_1, f_2, S, ϕ).

For the second ranking refinement strategy (i.e. iRANK-CRL), we need to compute the similarity matrix. It should be emphasized that the computation can be carried out offline to make iRANK-CRL efficient. We summarize the corresponding sentence ranking algorithms in Algorithm 2.

Algorithm 2. iRANK-CRL(f_1, f_2, D, q)

1. Extract sentences $S = \{s_1, \ldots s_m\}$ from D.
2. Calculate the similarity matrix, $W_{ij} = sim(s_i, s_j)$, $W_{ii} = 0$.
3. Calculate $\mathrm{H} = \Delta^{-1/2} W \Delta^{-1/2}$, Δ is a diagonal matrix with $\Delta_{ii} = \sum_j W_{ij}$.
4. Define the ranking refinement strategy,

$$\begin{cases} r_1^{(k+1)} = \phi(f_1, \tau) = (1-\beta)\cdot \mathrm{H}\cdot r_2^{(k)} + \beta \cdot r_1^{(*)} \\ r_2^{(k+1)} = \phi(f_2, \tau) = (1-\beta)\cdot \mathrm{H}\cdot r_1^{(k)} + \beta \cdot r_2^{(*)} \end{cases} \quad (15)$$

5. Return iRANK(f_1, f_2, S, ϕ).

Sentence Selection Strategy

The sentence selection strategy can indeed affect the quality of the system-generated summaries. However, because we focus on the sentence ranking in this article, we develop a simple yet effective sentence selection strategy as follows. We incrementally add into the summary the highest ranked sentence if it does not significantly repeat the information already included in the summary until the word limitation of the summary is reached. As in our experiments, a sentence is discarded if the cosine similarity of it to any sentence already selected into the summary is greater than 0.3. We find the cosine similarity among the top-ranked sentences, in most cases, is below 0.2 in our experiments, and we chose 0.3 as the threshold which is trained on DUC 2005 data set.

Experiment and Evaluation

Experiment Set-Up

We take the DUC 2005 and 2006 data set as the evaluation corpora. Table 1 below shows the basic statistics of the data sets. Each set of documents is accompanied with a query description representing a user's information need. The system-generated summaries are limited to 250 words in length.

The documents and the query descriptions are segmented into the sentences which are represented by the vectors of the words. The words are weighted by *tf-isf*, (as defined in a previous section). The stop-words in both documents and

TABLE 1. Basic statistics of the Document Understanding Conference (DUC) data sets.

	Total number of document sets	Average number of documents per set	Average number of sentences per set
DUC 2005	50	31.86	1002.54
DUC 2006	50	25	815.22

DOI: 10.1002/asi

queries are removed, and the remaining words are stemmed by Porter Stemmer.[3]

As for the evaluation metrics, ROUGE (recall-oriented understudy for gisting evaluation) 1.5.5 (Lin & Hovy, 2003), which is officially adopted in the DUC evaluations, is used in this study. ROUGE measures how well a machine summary overlaps with human summaries using N-gram co-occurrence statistics. Multiple ROUGE metrics are defined according to different N and different strategies, such as ROUGE-1 (Uni-gram based), ROUGE-2 (Bi-gram based), ROUGE-SU4 (skip-bi-gram based with maximum skip distance of 4, plus Uni-gram), and ROUGE-L (longest common subsequence based), etc. For the following experiments, we report the average recalls of ROUGE-1, ROUGE-2, and ROUGE-SU4, considering they are highly correlated with human judgments and have been taken as the official metrics in the DUC 2005 and 2006.

For the purpose of comparison, we implement the following two base rankers and the linear combination of them for reference.

> QRR: Query relevance-based ranker, which ranks the sentences according to their relevance to the query, i.e., the cosine similarity between the sentences and the query (i.e., f_1).
> LRR: LexRank-based ranker, which ranks the sentences using graph based ranking algorithm (i.e., f_2).
> LCR: Linear combined ranker, which linearly combines QRR and LRR. The linear combination parameter is denoted by λ, i.e., $LCR = QRR \cdot \lambda + (1 - \lambda) \cdot LRR$. Before the linear combination, both the QRR and the LRR scores are normalized by $\frac{(x - \min)}{(\max - \min)}$, where x denotes the original ranking score; max and min denote the maximum and minimum score values, respectively.

As for the termination condition, we set K (as noted in an earlier section) to 10 because 10 sentences are usually sufficient enough in the DUC query-focused summarization task.

Evaluation of Ranking Strategies

The aim of the first set of experiments is to compare the proposed rank-learn-combine approaches (i.e., iRANK) with the traditional "and-then-combine approach (i.e., LCR) and compare the different ranking refinement strategies (i.e., iRANK-FF vs. iRANK-CRL) on the DUC 2005 data set. The damping factor d in LRR is set to 0.75. To avoid the "link-by-chance" problem (i.e., the two sentences are linked together only because they share a word or two by chance), we set the values in the affinity matrix W to 0 if they are below a threshold 0.03. These parameters are tuned in our experiments. The settings of the other parameters are the combination factor $\lambda = 0.4$ in LCR, iRANK-FF, and iRANK-CRL; the balance factor $\eta = 0.8$ in iRANK-FF; and $\beta = 0.7$ in iRANK-CRL. We report the results on different λ in the next section, and different η and β in The Effect of Parameters section. Table 2 shows the results of average recalls of ROUGE-1, ROUGE-2, and ROUGE-SU4 along with their 95% confidence intervals included within the square brackets. Among them, ROUGE-2 is the primary DUC evaluation criterion.

TABLE 2. Compare different ranking strategies.

	ROUGE-1	ROUGE-2	ROUGE-SU4
QRR	0.3597	0.0664	0.1229
	(0.3540, 0.3654)	(0.0630, 0.0697)	(0.1196, 0.1261)
LRR	0.3679	0.0676	0.1234
	(0.3614, 0.3744)	(0.0643, 0.0708)	(0.1200, 0.1268)
LCR	0.3827	0.0776	0.1338
	(0.3766, 0.3884)	(0.0738, 0.0814)	(0.1301, 0.1375)
iRANK- FF	0.3878	0.0792	0.1366
	(0.3818, 0.3940)	(0.0754, 0.0832)	(0.1329, 0.1404)
iRANK-CRL	0.3880	0.0802	0.1373
	(0.3819, 0.3943)	(0.0763, 0.0841)	(0.1335, 0.1412)

TABLE 3. Improvement by the ranking refinement process.

	ROUGE-1	ROUGE-2	ROUGE-SU4
QRR	0.3597	0.0664	0.1229
	(0.3540, 0.3654)	(0.0630, 0.0697)	(0.1196, 0.1261)
QRR[a]	0.3631	0.0693	0.1255
	(0.3571, 0.3688)	(0.0656, 0.0730)	(0.1219, 0.1291)
QRR[b]	0.3720	0.0718	0.1285
	(0.3660, 0.3778)	(0.0680, 0.0756)	(0.1247, 0.1318)
LRR	0.3679	0.0676	0.1234
	(0.3614, 0.3744)	(0.0643, 0.0708)	(0.1200, 0.1268)
LRR[a]	0.3713	0.0692	0.1253
	(0.3644, 0.3781)	(0.0657, 0.0726)	(0.1217, 0.1289)
LRR[b]	0.3735	0.0726	0.1301
	(0.3673, 0.3792)	(0.0690, 0.0760)	(0.1264, 0.1340)

[a]Improvement by iRANK-FF. [b]Improvement by iRANK-CRL.

Notice that the improvement of LCR over QRR and LRR is rather significant if the value of the combination parameter λ is selected appropriately. It may imply that QRR and LRR indeed reflect different aspects of sentences. Besides, iRANK-FF and iRANK-CRL are superior to LCR. This is because both QRR and LRR are enhanced during ranking refinement, which, in turn, results in the increased overall performance. We will show the evidence in the next section. Moreover, the best result produced by iRANK-CRL[4] is better than that produced by iRANK-FF.

The Effect of Learning

The following experiments examine the improvement of the two base rankers through the ranking refinement. As shown in Table 3, both base rankers, i.e., QRR and LRR, are enhanced with the iRANK-FF and iRANK-CRL strategies, and the improvements are obvious. So, it is not surprising

[3]http://tartarus.org/~martin/PorterStemmer/

[4]We use the top 15% sentences as the feedback information at each round. We also find that the number of the sentences in the feedback information is not a sensitive factor in iRANK-CRL.

TABLE 4. LCR with different combination (λ) values.

λ	ROUGE-1	ROUGE-2	ROUGE-SU4
0.1	0.3742 (0.3674, 0.3806)	0.0704 (0.0669, 0.0739)	0.1269 (0.1233, 0.1305)
0.2	0.3796 (0.3734, 0.3856)	0.0728 (0.0692, 0.0764)	0.1296 (0.1261, 0.1331)
0.3	0.3836 (0.3775, 0.3898)	0.0760 (0.0720, 0.0799)	0.1333 (0.1295, 0.1371)
0.4	0.3827 (0.3766, 0.3884)	0.0776 (0.0738, 0.0814)	0.1338 (0.1301, 0.1375)
0.5	0.3798 (0.3732, 0.3863)	0.0773 (0.0733, 0.0812)	0.1331 (0.1293, 0.1369)
0.6	0.3774 (0.3713, 0.3831)	0.0758 (0.0721, 0.0795)	0.1324 (0.1287, 0.1360)
0.7	0.3736 (0.3677, 0.3791)	0.0742 (0.0704, 0.0779)	0.1305 (0.1269, 0.1341)
0.8	0.3698 (0.3639, 0.3753)	0.0727 (0.0691, 0.0763)	0.1283 (0.1249, 0.1316)
0.9	0.3651 (0.3592, 0.3706)	0.0699 (0.0663, 0.0734)	0.1258 (0.1225, 0.1292)

TABLE 5. iRANK-FF with different combination (λ) values.

λ	ROUGE-1	ROUGE-2	ROUGE-SU4
0.1	0.3775 (0.3710, 0.3837)	0.0720 (0.0686, 0.0754)	0.1289 (0.1254, 0.1324)
0.2	0.3796 (0.3734, 0.3857)	0.0728 (0.0692, 0.0763)	0.1305 (0.1270, 0.1339)
0.3	0.3843 (0.3781, 0.3903)	0.0773 (0.0735, 0.0812)	0.1345 (0.1307, 0.1384)
0.4	0.3878 (0.3818, 0.3940)	0.0792 (0.0754, 0.0832)	0.1366 (0.1329, 0.1404)
0.5	0.3833 (0.3773, 0.3893)	0.0789 (0.0752, 0.0828)	0.1347 (0.1310, 0.1384)
0.6	0.3792 (0.3732, 0.3851)	0.0776 (0.0738, 0.0814)	0.1333 (0.1296, 0.1370)
0.7	0.3746 (0.3686, 0.3805)	0.0748 (0.0708, 0.0787)	0.1312 (0.1275, 0.1349)
0.8	0.3709 (0.3651, 0.3765)	0.0730 (0.0694, 0.0768)	0.1293 (0.1258, 0.1328)
0.9	0.3684 (0.3626, 0.3740)	0.0721 (0.0686, 0.0759)	0.1279 (0.1245, 0.1313)

to see that the combinations of QRR and LRR (i.e., LCR) are also enhanced with iRANK in Table 2. In addition, the improvement by iRANK-CRL is more significant than the improvement by iRANK-FF. We thus come to the conclusion that iRANK-CRL is more effective than iRANK-FF.

Large-scale experiments are conducted to examine the effect of mutual refinement on the ranker combination. The combination results with/without involving any learning process are compared. Tables 4–6 present the ROUGE results of LCR, iRANK-FF, and iRANK-CRL with difference values of combination factor λ, respectively. In these experiments, the balance factor η in iRANK-FF is set to 0.8 in and the balance factor β in iRANK-CRL is set to 0.7. Let us compare the results in Tables 5 and 6 with those in Table 4. iRANK-FF and iRANK-CRL are always superior to LCR when the combination parameters of them are at the same level, ranging from 0.1 to 0.9. It suggests that mutual refinement can always improve the combination regardless of how the value of the combination parameter is selected. Again, iRANK-CRL is superior to iRANK-FF in all runs. These observations demonstrate the effectiveness of the ranking refinement, which further validates our motivation and the rationality of the iRANK framework as well as the two ranking refinement strategies.

TABLE 6. iRANK-CRL with different combination (λ) values.

λ	ROUGE-1	ROUGE-2	ROUGE-SU4
0.1	0.3788 (0.3725, 0.3851)	0.0723 (0.0687, 0.0756)	0.1293 (0.1259, 0.1327)
0.2	0.3802 (0.3738, 0.3866)	0.0736 (0.0700, 0.0771)	0.1308 (0.1272, 0.1343)
0.3	0.3839 (0.3776, 0.3901)	0.0775 (0.0735, 0.0813)	0.1341 (0.1303, 0.1340)
0.4	0.3880 (0.3819, 0.3943)	0.0802 (0.0763, 0.0841)	0.1373 (0.1335, 0.1412)
0.5	0.3871 (0.3809, 0.3932)	0.0798 (0.0760, 0.0837)	0.1368 (0.1331, 0.1408)
0.6	0.3863 (0.3803, 0.3927)	0.0792 (0.0754, 0.0831)	0.1363 (0.1325, 0.1402)
0.7	0.3767 (0.3705, 0.3828)	0.0753 (0.0714, 0.0792)	0.1317 (0.1280, 0.1354)
0.8	0.3731 (0.3671, 0.3789)	0.0742 (0.0705, 0.0780)	0.1305 (0.1269, 0.1341)
0.9	0.3695 (0.3635, 0.3753)	0.0732 (0.0695, 0.0771)	0.1230 (0.1264, 0.1335)

TABLE 7. iRANK-FF with different balance (η) values.

η	ROUGE-1	ROUGE-2	ROUGE-SU4
0.5	0.3790 (0.3727, 0.3853)	0.0775 (0.0736, 0.0816)	0.1332 (0.1294, 0.1370)
0.6	0.3858 (0.3797, 0.3918)	0.0787 (0.0747, 0.0826)	0.1357 (0.1319, 0.1395)
0.7	0.3870 (0.3808, 0.3932)	0.0793 (0.0755, 0.0832)	0.1361 (0.1324, 0.1399)
0.8	0.3878 (0.3818, 0.3940)	0.0792 (0.0754, 0.0832)	0.1366 (0.1329, 0.1404)
0.9	0.3866 (0.3804, 0.3925)	0.0791 (0.0751, 0.0831)	0.1363 (0.1323, 0.1402)

The Effect of Parameters

We then further examine the balance parameter settings in ranking refinement. Tables 7 and 8 show the results of iRANK-FF and iRANK-CRL with η and β ranging from 0.5 to 0.9. Notice that here η and β are not the combination factor as in LCR. We believe that a base ranker should have at least half belief in the ranking results of its own and thus the value of the balance factors η and β should be greater than 0.5. The combination factor λ is set to 0.4 according to the results obtained in previous experiments.

As shown in Tables 7 and 8, both iRANK-FF and iRANK-CRL produce relatively stable and promising results regardless of the change of η or β. iRANK-CRL is especially stable and the worst result produced by iRANK-CRL is

DOI: 10.1002/asi

TABLE 8. iRANK-CRL with different balance (β) values.

β	ROUGE-1	ROUGE-2	ROUGE-SU4
0.5	0.3876	0.0797	0.1370
	(0.3814, 0.3940)	(0.0758, 0.0836)	(0.1332, 0.1409)
0.6	0.3881	0.0799	0.1372
	(0.3819, 0.3945)	(0.0761, 0.0838)	(0.1335, 0.1411)
0.7	0.3880	0.0802	0.1373
	(0.3819, 0.3943)	(0.0763, 0.0841)	(0.1335, 0.1412)
0.8	0.3869	0.0798	0.1365
	(0.3807, 0.3929)	(0.0759, 0.0837)	(0.1327, 0.1403)
0.9	0.3864	0.0792	0.1361
	(0.3804, 0.3923)	(0.0753, 0.0831)	(0.1323, 0.1400)

TABLE 9. Compare different ranking strategies in Document Understanding Conference (DUC) 2006.

	ROUGE-1	ROUGE-2	ROUGE-SU4
QRR	0.3808	0.0785	0.1330
	(0.3755, 0.3862)	(0.0747, 0.0823)	(0.1297, 0.1364)
LRR	0.3943	0.0830	0.1383
	(0.3881, 0.4008)	(0.0787, 0.0875)	(0.1343, 0.1426)
LCR	0.3993	0.0889	0.1430
	(0.3940, 0.4046)	(0.0849, 0.0929)	(0.1397, 0.1465)
iRANK- FF	0.4016	0.0903	0.1446
	(0.3965, 0.4069)	(0.0863, 0.0942)	(0.1413, 0.1483)
iRANK-CRL	0.4032	0.0912	0.1450
	(0.3982, 0.4086)	(0.0871, 0.0956)	(0.1419, 0.1488)

TABLE 10. Compare with participating systems in Document Understanding Conference (DUC) 2005.

	ROUGE-1	ROUGE-2	ROUGE-SU4
H	–	0.0897	0.1510
iRANK-CRL		0.0802	0.1373
iRANK-FF	–	0.0792	0.1366
S15	–	0.0725	0.1316
S17	–	0.0717	0.1297
S10	–	0.0698	0.1253
S8		0.0696	0.1279
S4		0.0686	0.1277
NIST Baseline		0.0403	0.0872

Note. NIST = National Institute of National Standards and Technology

TABLE 11. Compare with participating systems in Document Understanding Conference (DUC) 2006.

	ROUGE-1	ROUGE-2	ROUGE-SU4
H	–	0.1036	0.1683
S24	–	0.0956	0.1553
iRANK-CRL	–	0.0912	0.1450
S15	–	0.0910	0.1473
iRANK-FF	–	0.0903	0.1446
S12	–	0.0898	0.1476
S8		0.0895	0.1460
S23		0.0879	0.1449
NIST Baseline		0.0495	0.0979

even better than the best result produced by iRANK-FF. This demonstrates that iRANK-CRL is more effective than iRANK-FF.

Evaluations on the DUC 2006 Data Set

We then conduct the follow-up experiments on DUC 2006 data set. We use the same parameters as described in a previous section, i.e., $\eta = 0.8$ in iRANK-FF, $\beta = 0.7$ in iRANK-CRL and $\lambda = 0.4$. As shown in Table 9 the improvements by iRANK-FF and iRANK-CRL are again visible.

Comparison With DUC Submissions

Finally, we compare our results with the DUC participating systems. To provide a global picture, we present the following representative ROUGE results of (a) the worst-scoring human summary (denoted by H), which reflects the margin between the machine-generated summaries and the human summaries; (b) the top five participating systems according to their ROUGE-2 scores (e.g., S15, S17, and so on); and (c) the National Institute of Standards and Technology (NIST) baseline, which forms its summaries by simply taking the first sentences in the documents until the summary length is achieved. Notice that ROUGE-1 scores and all the 95% confidential intervals are not officially released by DUC.

The advantage of iRANK is clearly shown in Tables 10 and 11. It produces very competitive results, which significantly outperform the NIST baselines in both years. More important, although iRANK-FF and iRANK-CRL are comparable to the top two systems in the DUC 2006, they are even superior to the best participating system in the DUC 2005. It follows that the application of the iRANK framework to DUC query-focused summarization is successful.

Conclusion

In this article, we propose a novel unsupervised ensemble ranking framework called interactive ranking (iRANK). We also design and investigate two ranking refinement strategies to use the feedback to support mutual learning between two base rankers so as to jointly improve the final overall ranking results. As a case study, we examine the proposed iRANK framework in the context of query-focused summarization. Encouraging results are achieved. However, there is still much room for improvement. In particular, we are interested in the following questions.

1. Does the quality of the two base rankers involved influence the final ranking results? Intuitively, the answer should be yes. Then another problem arises. Given a strong and a weak ranker, could the weak ranker enhance itself by learning from the strong one? How to prevent the strong ranker from getting weak when learning from the weak one? How could we ensure improved final ranking results? Although these concerns are easily understood and may be figured out using common sense, we need a mechanism to deal with them in the iRANK framework.

DOI: 10.1002/asi

2. How to extend iRANK from two rankers to n rankers? A simple yet possible method is to use iRANK with two rankers each time step by step, but is there any better choice available?
3. We would like to investigate other sound techniques to use as feedback and to think about how to extend the iRANK framework to other applications, such as opinion summarization where the integration of opinion-biased and document-biased ranking is necessary.

Last but not least, iRANK is not proposed as a new ranking algorithm, but a framework to better use the existing base rankers to effectively produce more reliable and accurate ranking results.

Acknowledgments

The first author and the third author would like to thank Michelle X. Zhou for her support in preparing an earlier draft of this manuscript. Work was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU5230/08E and PolyU5217/07E) and an internal grant from the Hong Kong Polytechnic University (Project No. A-PA6L).

References

Aslam, J.A., & Montague, M. (2001). Models for metasearch. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 276–284). New York: ACM Press.

Aslam, J.A., Pavlu, V., & Savell, R. (2007). A unified model for metasearch, pooling, and system evaluation. In Proceedings of the Twelfth International Conference on Information and Knowledge Management (pp. 484–491). New York: ACM Press.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory (pp. 92–100). New York: ACM Press.

Chen, S.C., Wang, F., Song, Y.Q., & Zhang, C.S. (2008). Semi-supervised ranking aggregation. In Proceedings of ACM Seventeenth Conference on Information and Knowledge Management (pp. 1247–1248). New York: ACM.

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the Web. In Proceedings of the Tenth International Conference on World Wide Web (pp. 613–622). New York: ACM Press.

Erkan, G., & Radev, D.R. (2004). LexRank: Graph-based centrality as salience in text summarization. Journal of Artificial Intelligence Research, 22, 457–479.

Fagin, R., Kumar, R., & Mahdian, M. (2004). Comparing and aggregating rankings with ties. In Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS) (pp. 47–58). New York: ACM Press.

Farah, M., & Vanderpooten, D. (2007). An outranking approach for rank aggregation in information retrieval. In Proceedings of the Thirtieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 591–598). New York: ACM Press.

Hoi, S.C.H., & Jin, R. (2008). Semi-supervised ensemble ranking. In Proceedings of the Twenty-third AAAI Conference on Artificial Intelligence (pp. 634–639). Menlo Park, CA: AAAI Press.

Jin, R., Valizadegan, H., & Li, H. (2008). Ranking refinement and its application to information retrieval. In Proceedings of the Seventeenth International Conference on World Wide Web (WWW) (pp. 397–406). New York: ACM Press.

Klementiev, A., Roth, D., & Small, K. (2008). Unsupervised rank aggregation with distance-based models. In Proceedings of the Twenty-fifth International Conference on Machine Learning (pp. 472–479). New York: ACM Press.

Li, W.J., Li, W., Li, B.L., Chen, Q., & Wu, M.L. (2005). The Hong Kong Polytechnic University at DUC 2005. In Proceedings of Document Understanding Conference 2005. Gaithersburg, MD: National Institute of Standards and Technology.

Lin, C.Y., & Hovy, E. (2000). The automated acquisition of topic signature for text summarization. In Proceedings of the Eighteenth International Conference on Computational Linguistics (pp. 495–501). College Park, MD: Association for Computational Linguistics.

Lin, C.Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of Human Language Technology conference/North American chapter of the Association for Computational Linguistics Annual Meeting (pp. 71–78). College Park, MD: Association for Computational Linguistics.

Liu, Y.T., Liu, T.Y., Qin, T., Ma, Z.M., & Li, H. (2007). Supervised rank aggregation. In Proceedings of the Sixteenth International Conference on World Wide Web (pp. 481–490). New York: ACM Press.

Montague, M., & Aslam, J.A. (2001). Relevance score normalization for metasearch. In Proceedings of ACM Tenth Conference on Information and Knowledge Management (pp. 427–433). New York: ACM Press.

Montague, M., & Aslam, J.A. (2002). Condorcet fusion for improved retrieval. In Proceedings of ACM Eleventh Conference on Information and Knowledge Management (pp. 538–548). New York: ACM Press.

Otterbacher, J., Erkan, G., & Radev, D.R. (2005, October). Using random walks for question-focused sentence retrieval. Paper presented at Human Language Technology Conference Conference on Empirical Methods in Natural Language Processing, Vancouver, Canada.

Ouyang, Y., Li, S.J., & Li, W.J. (2007). Developing learning strategies for topic-based summarization. In Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management (pp. 79–86). New York: ACM Press.

Over, P., Dang, H., & Harman, D. (2007). DUC in Context. Information Processing and Management, 43(6), 1506–1520.

Pickens, J., & Colovchinsky, G. (2008). Ranked feature fusion models for ad hoc retrieval. In Proceedings of ACM Seventeenth Conference on Information and Knowledge Management (pp. 893–900). New York: ACM Press.

Radev, D.R., Jing, H.Y., Stys, H., & Tam, D. (2004). Centroid-based summarization of multiple documents. Information Processing and Management, 40, 919–938.

Schilder, F., & Kondadadi, R. (2008). FastSum: Fast and accurate query-based multi-document summarization. In Proceedings of ACL-08 (pp. 205–228). College Park, MD: Association for Computational Linguistics.

Wan, X.J., & Yang, J.W. (2008). Multi-document summarization using cluster-based link analysis. In Proceedings of the Thirty-first Annual International ACM SIGIR Conference on Research and development in information retrieval (pp. 299–306). New York: ACM Press.

Wei, F.R., Li, W.J., Lu, Q., & He, Y.X. (2008). Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In Proceedings of the Thirty-first Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 283–290). New York: ACM Press.

Wong, K.F., Wu, M.L., & Li, W.J. (2008). Extractive summarization using supervised and semi-supervised learning. In Proceedings of the Twenty-second International Conference on Computational Linguistics (pp. 985–992). College Park, MD: Association for Computational Linguistics.

Zhou, D.Y., Bousquet, O., Lal, T.N., Weston, J., & Scholkopf, B. (2003). Learning with global and local consistency. In Advances in Neural Information Processing Systems (Vol. 16, pp. 321–328). Cambridge, MA: MIT Press.

DOI: 10.1002/asi

Appendix

Proof of iRANK-CRL Convergence

Lemma 1. The ranking results of iRANK-CRL as defined in Equation 9 converges.

Proof. Take r_1 for example, we hold, $r_1^{(i+1)} = \alpha \cdot \mathrm{H} \cdot r_2^{(i)} + \beta \cdot r_1^{(*)}$, then

$$\begin{aligned} r_1^{(i+1)} - r_1^{(i)} &= \alpha \cdot \mathrm{H} \cdot r_2^{(i)} + \beta \cdot r_1^{(*)} - \alpha \cdot \mathrm{H} \cdot r_2^{(i-1)} - \beta \cdot r_1^{(*)}, \\ &= \alpha \cdot \mathrm{H} \cdot (r_2^{(i)} - r_2^{(i-1)}) \end{aligned}$$

Since,
$r_2^{(i)} - r_2^{(i-1)} = \alpha \cdot \mathrm{H} \cdot (r_1^{(i-1)} - r_1^{(i-2)})$, we have,

$$\begin{aligned} r_1^{(i+1)} - r_1^{(i)} &= \alpha \cdot \mathrm{H} \cdot (\alpha \cdot \mathrm{H} \cdot (r_1^{(i-1)} - r_1^{(i-2)})) \\ &= \alpha^2 \cdot \mathrm{H}^2 \cdot (r_1^{(i-1)} - r_1^{(i-2)}) \end{aligned}$$

We have,

$$\begin{aligned} & r_1^{(i+1)} - r_1^{(i)} \\ &= \alpha^{i-j} \cdot \mathrm{H}^{i-j} \cdot (r_1^{(j+1)} - r_1^{(j)}), \quad \forall i \geq j \geq 0 \end{aligned}$$

So, we have,

$$\begin{aligned} r_1^{(i+1)} - r_1^{(i)} &= \alpha^i \cdot \mathrm{H}^i \cdot (r_1^{(1)} - r_1^{(0)}) \\ &= \alpha^i \cdot \mathrm{H}^i (\beta \cdot \mathrm{H} \cdot r_2^{(0)} + \alpha \cdot r_1^{(0)} - r_1^{(0)}) \\ &= \alpha^{i+1} \cdot \mathrm{H}^i \cdot (\mathrm{H} \cdot r_2^{(0)} - r_1^{(0)}) \\ &= \alpha^{i+1} \cdot \mathrm{H}^i \cdot (\mathrm{H} \cdot r_2^{(*)} - r_1^{(*)}) \end{aligned}$$

Since $0 < \alpha < 1$ and the eigenvalues of H in $[-1, 1]$ (H is similar to the stochastic matrix $\mathrm{P} = D^{-1}W = D^{-1/2}\mathrm{H}D^{1/2}$) (Zhou et al., 2004), therefore, $\lim_{i \to \infty} (\alpha^2 \mathrm{H}^2)^n = 0$.

Finally, we have,

$$\begin{aligned} \lim_{i \to \infty} |r_1^{(i+1)} - r_1^{(i)}| &= \lim_{i \to \infty} |\alpha^{i+1} \cdot \mathrm{H}^i \cdot (\mathrm{H} \cdot r_2^{(*)} - r_1^{(*)})| \\ &= 0 \end{aligned}$$

So, we can conclude that $r_1^{(i)}$ converges, and so $r_2^{(i)}$ does. □

DOI: 10.1002/asi