Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A ranking SVM based fusion model for cross-media meta-search engine*

Ya-li CAO[†1,2], Tie-jun HUANG[1,2], Yong-hong TIAN[1,2]
([1]Shenzhen Graduate School, Peking University, Shenzhen 518055, China)
([2]Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)
[†]E-mail: ylcao@jdl.ac.cn

Received Sept. 14, 2010; Revision accepted Oct. 9, 2010; Crosschecked Sept. 26, 2010

Abstract: Recently, we designed a new experimental system MSearch, which is a cross-media meta-search system built on the database of the WikipediaMM task of ImageCLEF 2008. For a meta-search engine, the kernel problem is how to merge the results from multiple member search engines and provide a more effective rank list. This paper deals with a novel fusion model employing supervised learning. Our fusion model employs ranking SVM in training the fusion weight for each member search engine. We assume the fusion weight of each member search engine as a feature of a result document returned by the meta-search engine. For a returned result document, we first build a feature vector to represent the document, and set the value of each feature as the document's score returned by the corresponding member search engine. Then we construct a training set from the documents returned from the meta-search engine to learn the fusion parameter. Finally, we use the linear fusion model based on the overlap set to merge the results set. Experimental results show that our approach significantly improves the performance of the cross-media meta-search (MSearch) and outperforms many of the existing fusion methods.

Key words: Information fusion, Meta-search, Cross-media, Ranking
doi:10.1631/jzus.C1001009 **Document code:** A **CLC number:** TP391

1 Introduction

In addition to text-based retrieval for books, image retrieval is also an important part of the development of the Universal Digital Library (UDL). We first designed a visual, hierarchical e-book browsing and retrieval system, KnowMap, based on a topic map. Recently, we designed another image retrieval system, MSearch, based on the image database of the WikipediaMM task of ImageCLEF 2008 (Zhou *et al*., 2008). MSearch is a cross-media meta-search engine using both the text and visual features of an image in the retrieval process.

A meta-search engine (Aslam and Montague, 2001) is an information retrieval agent built on top of other search engines. Selberg and Etzioni (1995) designed the first meta-search engine Meta Crawler. The meta-search engine sends users' requests (queries) to several member search engines, and then aggregates the results into one result list. The kernel problem is how to merge the results from multiple member search engines and provide a better rank list. A good results fusion method can provide more comprehensive and precise information to users. To enhance the performance of the meta-search engine, there exist many fusion methods, such as the Borda count (BC) model (van Erp and Schomaker, 2000; Aslam and Montague, 2001; Dwork *et al*., 2001), the comb model (Fox and Shaw, 1993), and the round robin (RR) model (Cao *et al*., 2009). In this paper, we propose a novel fusion method for meta-search engines.

Nowadays there are a large number of meta-search engines on the Internet; however, most of them are text-based meta-search engines. For a given

* Project supported by the National Natural Science Foundation of China (No. 60605020) and the National High-Tech R & D Program (863) of China (Nos. 2006AA01Z320 and 2006AA010105)

meta-search engine, although the retrieval algorithms that its member search engines employ are different, the text-based techniques are much more mature. That is, the performances of the member search engines are almost the same. Our meta-search engine is a cross-media engine including one text-based and one content-based member system. Although both of our retrieval approaches show good performance in image retrieval, the two subsystems' performances are quite different. In the face of a meta-search engine different from the traditional ones, we propose a novel results fusion model based on ranking support vector machine (SVM).

The contributions of this paper include:

1. We proposed a supervised learning approach employing the supervised learning approach.

2. We proposed an application of Ranking SVM for meta-search engine, and finally transformed the fusion problem into an optimization problem.

3. We carried out groups of experiments and verified the effectiveness of the proposed method.

2 Related works

2.1 Results fusion models

The goal of results fusion, sometimes called 'rank aggregation', is to combine the results from multiple ranking lists and generate a better ranking list. Typically there are two categories of results fusion methods, the score-based fusion method and the order-based fusion method. Whether a fusion method is score- or order-based depends on whether we can obtain the scores or the order of the results in the ranking list. In this work, our proposed fusion method is a score-based one.

In the past few years, researchers have paid considerable attention to the results fusion method. Fox and Shaw (1993) proposed a fusion method based on the min, max, median, or sum of each of the normalized relevance score of the member systems, which is overall called the comb model. The Borda count (BC) model is another well-known results fusion model, used for voting at the beginning; it sorts the results based on their position in the ranking lists. For example, given any query, a returned document is sorted according to the number of documents that are ranked below it in all the ranking lists. The RR model is another classical fusion model used for the meta-search engine. The idea of the RR model is very simple. For a meta-search engine, we first array the member search engines in some order, and then display the first item of each ranking list one by one, then the second item of each ranking list, and so forth. Recently, more fusion models have evolved from the classic models, these models including the median rank (Fagin *et al.*, 2003), fuzzy logic based fusion model (Ahmad and Sufyan Beg, 2002), genetic algorithm (Sufyan Beg, 2004), and position and snippets/titles based fusion model (Yuan and Wang, 2009). However, these results fusion models are mainly without supervised learning, in the sense that no training data is used. In addition, these fusion models are employed mainly as the fusion strategy of text-based meta-search engines. That is to say, the fusion method for content-based retrieval systems such as cross-media meta-search engines remains an open issue. For content-based retrieval systems, researchers care more about how to combine the text- and content-based methods together to integrate a single retrieval system, which looks like a vertical combination. In this study, our proposed method is more like a horizontal combination of the text- and content-based methods.

As is known, content-based retrieval systems usually do not perform as well as the text-based retrieval systems; that is, one cannot treat ranking lists from member search engines of different types equally. However, the results fusion methods we have discussed above assign the same weight for each member search engine. Motivated by this, we argue that in order for enhancing the accuracy of the results fusion process, it is better to employ a supervised learning approach to learn the difference between the performances among retrieval systems. Compared with the unsupervised fusion methods, there are several advantages for taking the supervised learning method. First, we make full use of the information existing in the labeled training data and the users' feedback. Second, we transform the fusion problem to a classification problem. There are many mature optimization techniques for obtaining the best fusion weight for each member search engine. Certainly, the supervised learning method has its own disadvantages. Employing this method may take much time in labeling the training set or extracting information from the users' feedback, and the problem caused by learning should be considered as another important issue to focus on in the future study.

2.2 ImageCLEF2008 and MSearch

ImageCLEF2008 is the cross-language image retrieval track run as part of the Cross-Language Evaluation Forum (CLEF) campaign. This track evaluates the retrieval of images described by text captions based on queries in a different language; both text and image matching techniques are potentially exploitable. In this competition, our text-based image retrieval (TBIR) approach ranked the first place among all submitted runs (Zhou *et al.*, 2008). Although not submitted, we then proposed a content-based image retrieval (CBIR) approach, which performed better than the other submitted CBIR runs in ImageCLEF2008.

Based on the two retrieval approaches, we designed a cross-media meta-search engine, MSearch, by combing the text- and content-based systems. MSearch provides not only normal text- or content-based retrieval functions, but also a meta-search function. Fig.1 shows the user interface of our cross-media meta-search engine.

Fig. 1 User interface of our cross-media meta-search engine, MSearch

As shown in Fig. 1, for meta-search users can submit text first, and then the cross-media system will present users the candidate images. In the next step, users can select any image from the recommended images as a query sample, and click the button 'MSearch'. Finally, our cross-media meta-search engine will present the users a results list by combining the results returned by both member search engines.

Take 'flower' for example. The query interface is as shown in Fig. 1. Users first input the text query 'flower', and then the system will return some recommended candidate image samples. Users can select any one as the image sample query and submit it to the retrieval system. Then MSearch deals with both of the queries and returns the meta-search results (Fig. 2).

Fig. 2 Results returned by MSearch with query 'flower'
(a) By cross-media retrieval approach; (b) By text-based approach; (c) By content-based approach

MSearch includes four main functional modules which work together to generate the final retrieval results (Zhou *et al.*, 2008):

1. Data processing module: a processing unit that performs several pre-processing tasks for the queries and the dataset with textual queries and then returns relevant images.

2. Text-based retrieval module: a retrieval subsystem that searches the dataset with textual queries and then returns relevant images.

3. Content-based retrieval module: a retrieval subsystem that searches the dataset with visual features and then returns relevant images.

4. Cross-media re-ranking module: a processing unit that combines the sets of returned images from CBIR and text-based retrieval modules, and then performs cross-media re-ranking to obtain the final retrieval results.

Fig. 3 shows the architecture of MSearch.

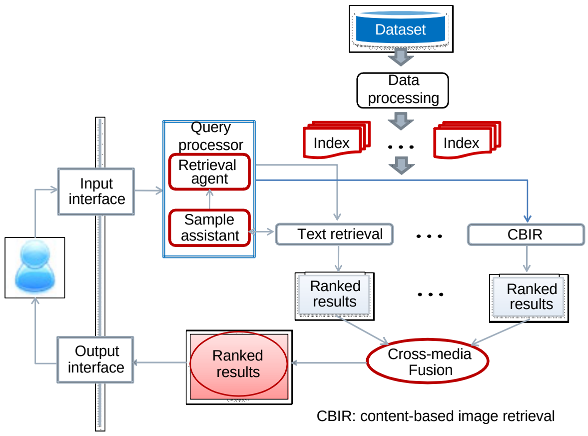

Fig. 3 Architecture of MSearch for the WikipediaMM 2008 task

The cross-media characteristics of our meta-search engine are shown in the following two aspects: (1) Users can obtain different types of media information using a query of a single media type; for example, people can use text to search for images, or use image samples to search for videos. (2) The retrieval systems can use different kinds of media features to fuse the final results. In our system, users can submit text or image samples to retrieval information. Then they can obtain an aggregated ranking list after dealing with all the text features and visual features.

3 Methods

In this section, we first introduce the basic principle of ranking SVM and then describe our fusion method in detail.

As discussed before, a meta-search engine is aimed to find an effective fusion method to merge the results from the member search engines and provide the user a better ranking list. However, a common issue exists in that the order- or score-based information obtained from member search engines cannot be compared directly. There are two main factors related to the issue: (1) The scores returned by different search engines have different baselines as different algorithms are used to generate the results' scores. (2) The retrieval algorithms are different for different search engines, leading to different retrieval performances for different search systems; that is, some search systems may be superior to others. If we do not take this issue into account, we may obtain an aggregated list which offers quantity rather than quality; that is, the precision of the aggregated list may be lower than that of the best member search engine.

To solve this problem, an alternative is to allocate different fusion weights to the member search engines. However, most of the unsupervised fusion methods treat all the ranking lists generated from different search engines equally. In our results fusion model, we take the performances of the member search engines into consideration. We use a supervised fusion method based on the ranking SVM method to learn the fusion weights for each member search engine. We carried out experiments on our cross-media meta-search engine MSearch whose member search engine performances are quite different. Experimental results show that our method outperforms the other unsupervised methods and enhances the performance of the cross-media meta-search engine compared with its member search systems.

3.1 Ranking SVM algorithm

Ranking is the key problem for information retrieval and other text applications. Recently, the learning-to-rank method or machine-learned ranking (MLR) has become a research focus. Learning-to-rank (Liu, 2009) is a type of supervised or semi-supervised machine learning problem aimed to automatically construct a ranking model from training data. Training data consists of lists of items with some partial order specified between items in each list. This order is typically induced by giving a numerical or ordinal score or a binary judgment (e.g., 'relevant' or 'irrelevant') for each item. The ranking SVM algorithm is a typical learning-to-rank method. Herbrich *et al.* (2000) first applied a large margin to ranking and formed the primary frame for ranking SVM. Then Joachims (2002) proposed the ranking SVM algorithm using implicit relevance feedback.

The basic idea of ranking SVM is to formalize learning-to-rank as a problem of binary classification on instance pairs, and then to solve the problem using an SVM. The process of ranking SVM includes mainly two steps. In the first step, a function $f(x)$ is used to transform the instance vector into the real number. For simplicity, a linear function is usually chosen to represent an instance vector. Here we define the function as

$$f(\boldsymbol{c},q)=\boldsymbol{w}\cdot h(\boldsymbol{c},q)\,, \tag{1}$$

where q is the query, $\boldsymbol{c}$ is the initial instance vector, and $\boldsymbol{w}$ is a weight vector used for the transformation.

For a query q_k, if an instance $\boldsymbol{c}_i$ is ranked higher than $\boldsymbol{c}_j$, i.e., $\boldsymbol{c}_i \succ \boldsymbol{c}_j$, the formula is denoted as

$$\begin{aligned}\boldsymbol{c}_i \succ \boldsymbol{c}_j \Leftrightarrow\ & g(\boldsymbol{c}_i,\boldsymbol{c}_j,q_k)\\ &= f(\boldsymbol{c}_i,q_k)-f(\boldsymbol{c}_j,q_k)\\ &=\boldsymbol{w}\cdot(h(\boldsymbol{c}_i,q)-h(\boldsymbol{c}_j,q))>0.\end{aligned} \tag{2}$$

For a ranking list with n instances, if we know the preference order of the n instances, then the ranking problem with n instances can be transformed to a binary classification problem with n^2 pairs of instances.

The second step of ranking SVM is to construct the SVM model for solving the binary classification problem. The quadratic convex optimization problem in ranking SVM is defined as (Joachims, 2002)

$$\text{minimize } V(\boldsymbol{w},\boldsymbol{\xi})=\frac{1}{2}\boldsymbol{w}\cdot\boldsymbol{w}+C\sum\xi_{i,j,k} \tag{3}$$

subject to

$$\forall i,j,k,\ \boldsymbol{w}\cdot(h(\boldsymbol{c}_i,q_k)-h(\boldsymbol{c}_j,q_k))\geq 1-\xi_{i,j,k}, \tag{4}$$

where C is a parameter that allows the trade-off between the margin size and the training error.

For a ranking SVM, the task of learning ranking function is not completely the same as that of learning classification function. There are two points to which we have to pay attention (Yu and Kim, 2010):

1. In ranking, a training set is an ordering of data. Let 'A is preferred to B' be specified as '$A\succ B$'. A training set for ranking SVM is denoted as

$$R=\{(x_i,y_i),\ (x_{i+1},y_{i+1}),\ ...,\ (x_m,y_m)\},$$

where y_i is the ranking of x_i; that is, $y_i<y_j$, if $x_i\succ x_j$.

2. A ranking function outputs a score for each data instance, from which a global ordering of data is constructed. That is, the target function $f(x_i)$ outputs a score such that $f(x_i)>f(x_j)$ for any $x_i\succ x_j$.

Above all, the key point of constructing a ranking SVM is to build the training set based on users' preferences and obtain the constraint relationships between the candidate instances. After that, we can minimize the loss function according to the constraint relationships and obtain the training parameters.

3.2 Fusion model based on ranking SVM

The ranking SVM algorithm is one of the classical learning-to-rank methods and has many applications in information retrieval, such as document retrieval, collaborative filtering, and sentiment analysis.

Ranking SVM has two main functions, prediction and sorting. In the prediction phase, ranking SVM receives the training data of multiple features, and then outputs an estimated weight for each feature. In the sorting phase, for an instance in the testing set, ranking SVM provides various fusion models to combine the scores of its features and outputs the final score of the instance.

In our proposed method, we employ the prediction function of ranking SVM to learn the fusion weight for the meta-search engine. To build ranking SVM, we should prepare for two things: (1) feature selection, and (2) the preference of training data. In our fusion model, we use the fusion weights of the member search engines to build the feature vector of an instance and use the labeled ground truth as the training set. The specific details in our algorithm are described as follows.

Assume there is a meta-search engine denoted as MSE, which has n member search engines, denoted as SE_1, SE_2, …, SE_n.

For a query q, each member search engine of the meta-search engine returns its ranking list, denoted as $\{\text{rank}_1, \text{rank}_2, …, \text{rank}_n\}$. We define the results set G of the meta-search engine for query q as $G=\{\text{rank}_1\}\cup\{\text{rank}_2\}\cup…\cup\{\text{rank}_n\}$.

For any returned document $d\in G$, we assign n features for it to build the feature vector of the documents. Each feature represents the retrieval performance of one member search engine. Then we set the value of each feature as the score of the document d obtained in the corresponding member search engine.

$$d=\{\text{feature}_1, \text{feature}_2, \ldots, \text{feature}_n\} \\ =\{\text{score}_1, \text{score}_2, \ldots, \text{score}_n\}. \quad (5)$$

According to manual annotation or users' feedback, we obtain the target order of all the returned documents in the training set. Then we use the ranking SVM method to learn the weight of each feature, denoted as $\{\text{wt}_1, \text{wt}_2, \ldots, \text{wt}_n\}$. Thus, the output of the prediction process of ranking SVM is the fusion weight for each member search engine.

Finally, we employ the linear fusion model to aggregate all the returned ranking lists and output a new ranking list to the user. The final aggregate function is

$$\text{score}_{\text{final}}(d)=\sum_{i=1}^{n}\text{score}_i\cdot\text{wt}_i. \quad (6)$$

For example, assume there is a meta-search engine having five member search engines A, B, C, D, E. For a query q, A, B, C, D, E each returns its retrieved results. d_1, d_2, d_3, d_4 are all of the documents belonging to the training set. The scores of each document returned by the five member search engines are shown as

$$d_1: \{1, 1, 0, 0.2, 0\};\ d_2: \{0, 0, 1, 0.1, 1\}; \\ d_3: \{0, 1, 0, 0.4, 0\};\ d_4: \{0, 0, 1, 0.3, 0\}.$$

And the target order of the four documents is {3, 2, 1, 1}, which indicates that

$$d_1 \succ d_2,\ d_1 \succ d_3,\ d_1 \succ d_4,\ d_2 \succ d_3,\ d_2 \succ d_4.$$

Then we use the ranking SVM algorithm to train the weights for the five features, i.e., the fusion weights of each member search engine, and obtain the following results:

FusionWeight(A): wt_1=0.30000001,
FusionWeight(B): wt_2=0.1,
FusionWeight(C): wt_3=−0.1,
FusionWeight(D): wt_4=−0.070000008,
FusionWeight(E): wt_5=0.1.

Given one instance of the testing set, d_i: {1, 1, 1, 1, 1}, the final score of d_i is computed as

$$\text{score}(d_i)=\text{wt}_1\times1+\text{wt}_2\times1+\text{wt}_3\times1+\text{wt}_4\times1+\text{wt}_5\times1 \\ =0.30000001+0.1-0.1-0.070000008+0.1 \\ =0.330000002.$$

4 Experiments

The following experiments verify whether the SVM fusion model can successfully be applied to the meta-search engine. The experiments were based on the framework of MSearch, which has two member search engines, one being a text-based information retrieval (TBIR) system, and the other a content-based information retrieval (CBIR) system.

4.1 Database

The experiments were carried out on the database of the WikipediaMM2008 task. In ImageCLEF, our text-based retrieval method won the first place of all the teams participating in the contest. The database includes about 150 000 pictures. The ground-truth results are given in the evaluation phase of the Wikipedia task including 75 topic queries and the relative pictures corresponding to the queries. In our experiments, we randomly split the ground truth into two groups, one used as the training set including pictures covering 35 topic queries and the other as the testing set including the other pictures covering the other 40 topic queries.

4.2 Evaluation measurements

Reasonable evaluation measurements can help to improve the performance of the retrieval system. In our experiments, we apply precision, recall, $P@N$, MAP (mean average precision), and R-precision as the evaluation measurements.

Precision and recall are two widely used statistical measurements. Precision can be seen as a measurement of exactness of fidelity, whereas recall is a measurement of completeness. In information retrieval, precision is defined as the ratio of the number of relevant documents retrieved by a search to the total number of documents retrieved by that search, and recall is defined as the ratio of the number of relevant documents retrieved by a search to the total number of existing relevant documents (which should have been retrieved).

P@N is the precision of the top *N* returned documents, defined as

$$P@N = \frac{1}{N}\sum_{i=1}^{N}\mathrm{rel}(d_i), \tag{7}$$

where

$$\mathrm{rel}(d_i) = \begin{cases} 1, & \text{if the document is relevant,} \\ 0, & \text{otherwise.} \end{cases}$$

MAP is the mean average precision over all queries, defined as

$$\mathrm{MAP} = \frac{1}{m}\sum_{j=1}^{m}\mathrm{AP}_j = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{R_j}\left(\sum_{i=1}^{k}\mathrm{rel}_j(d_j)\cdot(P@i)_j\right), \tag{8}$$

where m is the total number of the queries, R_j is the number of the relevant documents for the jth query, k is the number of the returned documents for the query, and $\mathrm{rel}_j(d_j)$ and $(P@i)_j$ are the values of $\mathrm{rel}(d_j)$ and $P@i$ for the jth query respectively.

R-precision is precision at cutoff *R*, PC(*R*), where *R* it the total number of relevant documents for the query. PC(*R*) implicitly assigns a weight of 1/*R* to each of the top *R* documents in a list and a weight of 0 to every remaining document.

4.3 Experimental results

The experiments were used to evaluate the performance of our proposed fusion model for meta-search engines. We implemented the three fusion models as discussed in Section 2. When training the ranking SVM, no kernel was used, and the trade-off between the training error and the margin was selected from $C\in\{0.01, 0.03, 0.05, 0.10\}$ by minimizing the leave-one-out error on the training set.

Table 1 shows the predictive fusion weight of each member search engine. The performance of TBIR was better than that of CBIR as TBIR gained the larger fusion weight.

Table 1 Fusion weight of each member search

C	Fusion weight	
	TBIR	CBIR
0.10	4.5111880	3.2298124
0.05	3.6797829	2.9072418
0.03	3.3114314	2.4794912
0.01	2.1483867	1.6006933

TBIR: text-based information retrieval; CBIR: content-based information retrieval

Table 2 lists the performance in many measures of the two retrieval systems. The text-based approach performed better than the content-based approach, meaning that our fusion weights for the two search engines are reasonable.

Table 2 Experimental results on WikipediaMM2008

Run ID	MAP	*P*@5	*P*@10	*P*@20
RSVM, *C*=0.10	**0.3737**	**0.6200**	0.5025	0.3150
RSVM, *C*=0.05	0.3733	**0.6200**	0.5025	**0.3175**
RSVM, *C*=0.03	0.3734	**0.6200**	0.5025	**0.3175**
RSVM, *C*=0.01	0.3722	0.6150	0.5000	0.3000
Borda	0.3696	0.6000	**0.5075**	0.3000
CombSum	0.2252	0.3000	0.3375	0.2200
CombANZ	0.2255	0.3100	0.3325	0.2250
RoundRobin	0.3172	0.5250	0.4550	0.2850
Text	0.3363	0.5450	0.4625	0.2425
Cbir	0.2421	0.5350	0.4275	0.1725
Run ID	***P*@30**	***P*@40**	***P*@5**	***R*-precision**
RSVM, *C*=0.10	**0.1800**	0.1525	**0.1350**	**0.3693**
RSVM, *C*=0.05	0.1675	0.1600	0.1325	0.3691
RSVM, *C*=0.03	0.1750	0.1550	**0.1350**	0.3690
RSVM, *C*=0.01	0.1700	**0.1625**	0.1175	0.3743
Borda	0.2050	0.1350	0.1575	0.3702
CombSum	0.1950	0.1550	0.1300	0.2479
CombANZ	0.1925	0.1575	0.1300	0.2481
RoundRobin	0.1825	0.1400	0.1500	0.3285
text	0.1925	0.1375	0.1225	0.3527
cbir	0.0925	0.0800	0.0600	0.2763

RSVM: ranking SVM. MAP: mean average precision

Table 2 shows the experimental results of fusion methods used for our cross-media meta-search system. For a meta-search engine, the efficiency of the results fusion method directly decides the final performance of a meta-search engine. If the fusion method does not work well, the results fusion for a meta-search engine cannot improve the retrieval performance; instead, it may lower the retrieval performance of a meta-search engine. For a meta-search engine whose member search engines have different performances, taking into account the fusion weights of the member search engines is necessary. Table 2 also shows than the Comb methods and RR methods did not improve the performance of the meta-search engines by emerging the results set while the other method played a positive role in improving the performance of a meta-search engine.

Additionally, Table 2 shows that our fusion method outperformed all the other fusion methods in several evaluation measurements. All of our four SVM fusion runs had higher MAP and *R*-precision values than the other fusion runs. In other evaluations, our methods also showed a better performance. Except for the *P*@10, our four fusion methods achieved the best performance in *P*@*N*. Especially, the SVM fusion run with *C*=0.10 showed the best performance; it improved the MAP value from 0.3363 to 0.3737, 11.1% higher than that of the text-based retrieval system. Moreover, it showed the best *R*-precision value as well, 4.7% higher than that of the text-based retrieval system.

Fig. 4 is the precision-recall graph of the final results generated by all experimental fusion methods, showing that our method achieved the best performance.

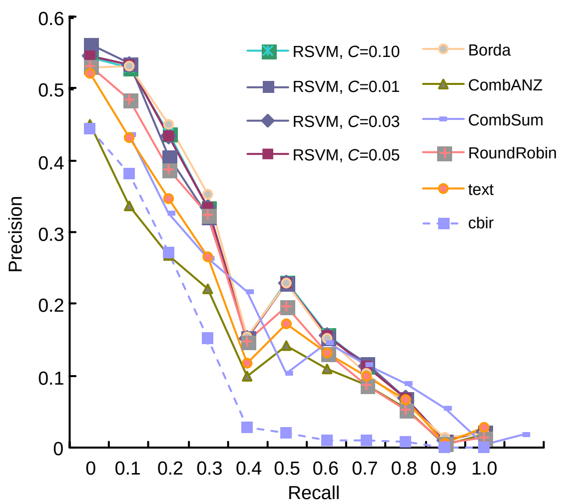

Fig. 4 The precision-recall graph of the final results generated by each fusion method
RSVM: ranking SVM

5 Conclusions

We propose a novel results fusion model based on ranking SVM, taking into account both text and visual features. We use the ranking SVM to generate an estimated fusion weight for a meta-search engine, and a linear model to compute the final score of the returned documents. Result of experiments carried out on the WikipediaMM database showed that the proposed method outperforms traditional fusion methods in terms of MAP, *P*@*N*, *R*-precision, and other evaluation measures. In the future, we will make further efforts to design a more effective fusion model at low cost.

References

Ahmad, N., Sufyan Beg, M.M., 2002. Fuzzy Logic Based Rank Aggregation Methods for the World Wide Web. Int. Conf. on Arifical Intelligence in Engineering and Technology, p.363-368.

Aslam, J.A., Montague, M., 2001. Models for Metasearch. Proc. 24th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.276-284. [doi:10.1145/383952.384007]

Cao, L., Han, L.X., Wu, S.L., 2009. Ranking algorithm for meta-search engine. *Appl. Res. Comput.*, **26**(2):411-414 (in Chinese).

Dwork, C., Kumar, R., Naor, M., Sivakumar, D., 2001. Rank Aggregation Methods for the Web. 10th Int. World Wide Web Conf., p.613-622. [doi:10.1145/371920.372165]

Fagin, R., Kumar, R., Sivakumar, D., 2003. Efficient Similiarity Search and Classification via Rank Aggregation. Proc. ACM SIGMOD Int. Conf. on Management of Data, p.301-312. [doi:10.1145/872757.872795]

Fox, E.A., Shaw, J.A., 1993. Combination of Multiple Searches. The Text Retrieval Conf., p.243-252.

Herbrich, R., Graepel, T., Obermaye, K., 2000. Large Margin Rank Boundaries for Ordinal Regression. Advances in Large Margin Classifiers, p.115-132.

Joachims, T., 2002. Optimizing Search Engines Using Clickthrough Data. Proc. ACM Conf. on Knowledge Discovery and Data Mining (KDD), p.133-142. [doi:10.1145/775047.775067]

Liu, T.Y., 2009. Learning to ranking for information retrieval. *Found. Trends Inf. Retr.*, **3**(3):225-331. [doi:10.1561/1500000016]

Selberg, E., Etzioni, O, 1995. Multi-Service Search and Comparison Using the Metacrawler. The 4th World Wide Web Conf., p.195-208.

Sufyan Beg, M.M., 2004. Parralel Rank Aggregation for the World Wide. Intelligent Sensing and Information Processing, p.385-390. [doi:10.1109/ICISIP.2004.1287688]

van Erp, M., Schomaker, L., 2000. Variants of the Borda Count Method for Combining Ranked Classifier Hypotheses. 7th Int. Workshop on Frontiers in Handwriting Recognition, p.443-452.

Yu, H., Kim, S., 2010. SVM Turorial: Classification, Regression, and Ranking. *In*: Handbook of Natural Computing. Springer.

Yuan, F.Y., Wang, J.D., 2009. An Implemented Rank Merging Algorithm for Meta Search Engine. Research Challenges in Computer Science, p.191-193. [doi:10.1109/ICRCCS.2009.56]

Zhou, Z., Tian, Y.H., Li, Y.N., Liu, T., Huang, T.J., Gao, W., 2008. PKU at ImageCLEF 2008: Experiments with Query Extension Techniques for Text-Based and Content-Based Image Retrieval. Online Working Notes for the CLEF Workshop.

Zhou, Z., Tian, Y.H., Li, Y.N., Huang, T.J., Gao, W., 2009. Large-Scale Cross-Media Retrieval of WikipediaMM Images with Textual and Visual Query Expansion. Cross-Language Evaluation Forum, p.763-770. [doi:10.1007/978-3-642-04447-2_99]