Inverse square rank fusion for multimodal search

Conference Paper · June 2014

DOI: 10.1109/CBMI.2014.6849825

3 authors:

André Mourão
Universidade NOVA de Lisboa

Flávio Martins
Universidade NOVA de Lisboa

João Magalhães
Universidade NOVA de Lisboa

Inverse Square Rank Fusion for Multimodal Search

André Mourão
Dep. Computer Science
Universidade Nova Lisboa
Portugal
a.mourao@campus.fct.unl.pt

Flávio Martins
Dep. Computer Science
Universidade Nova Lisboa
Portugal
flaviomartins@acm.org

João Magalhães
Dep. Computer Science
Universidade Nova Lisboa
Portugal
jm.magalhaes@fct.unl.pt

***Abstract*—Rank fusion is the task of combining multiple ranked document lists (ranks) into a single ranked list. It is a late fusion approach designed to improve the rankings produced by individual systems. Rank fusion techniques have been applied throughout multiple domains: e.g. combining results from multiple retrieval functions, or multimodal search where several feature spaces are common. In this paper, we present the Inverse Square Rank fusion method family, a set of novel fully unsupervised rank fusion methods based on quadratic decay and on logarithmic document frequency normalization. Our experiments created with standard Information Retrieval datasets (image and text fusion) and image datasets (image features fusion), show that ISR outperforms existing rank fusion algorithms. Thus, the proposed technique has comparable or better performance than existing state-of-the-art approaches, while maintaining a low computational complexity and avoiding the need for document scores or training data.**

I. Introduction

Rank fusion aims at combining ranked document lists (ranks) from multiple sources into a single (combined) ranked list. These techniques have been applied to solve multiple problems: combining results from multiple textual retrieval functions [1], combining the results of multimodal queries, federated search and expert search [2], among others. Unsupervised methods like score-based fusion [3] and rank-based fusion [4], [1] achieve good results without the need for document specific techniques or document scores. Alternative supervised methods like learning-to-rank [5] or boosting algorithms [6], [7] require a training phase with specific data and have a higher computational complexity.

Existing approaches for rank fusion are based on the rank or score of the documents across ranks. These approaches follow the effects described by Vogt et Cottrell [8]. They argue that there are three effects that can and are explored by rank fusion algorithms:

- The Skimming Effect: different retrieval approaches retrieve different relevant items
- The Chorus Effect: several retrieval approaches retrieve the same item
- The Dark Horse Effect: some retrieval approaches may produce more (or less) accurate estimates of relevance for some items, relative to other approaches

The exploitation of these effects is very clear in rank fusion approaches: for example, rank fusion algorithms like Reciprocal Rank [4] and Reciprocal Rank Fusion [1] work by giving fusion scores to documents by summing the inverse of the document rank in the individual ranks. This can be mapped to the Skimming Effect (retrieve top-ranked documents for each retrieval approach). The Chorus effect is visible in combMNZ [3]; the final sum of document score is multiplied by the number of engines where a document appears. The Dark Horse Effect is visible on the weighed (supervised) versions of CombSUM (wCombSUM) by multiplying the score of an engine to a certain query or query category. The Dark Horse effect relies on the relation between query and the search engines, meaning that it requires training data to be exploited. Although this paper is focused on unsupervised variants of the algorithm, this technique can also be easily incorporated in the proposed approach. See section III for more details. We will focus on the chorus and skimming effects. These effects are not exclusive and exploiting them may produce dissonant results (e.g. a large chorus effect cuts the possible gain from the skimming effect). Thus, one must be careful when exploring these effects.

We believe that existing rank fusion techniques can be improved by better exploiting these effects and adapt better to search engines. Inspired by the described effects and existing techniques, we propose the Inverse Square Rank fusion method, based on the quadratic decay of the document rank position and logarithmic document-frequency normalization. Evaluation on a multimodal medical dataset (text and images) and image search (multiple features) shows that ISR outperformed existing state-of-art methods, including the Reciprocal Rank [4], Reciprocal Rank Fusion [1] and score-based approaches [3]. ISR method is very fast and require no tuning or training, which means it can be used in real time and readily deployed.

This paper is organized as follows: section 2 reviews the state-of-the-art in rank fusion methods and section 3 presents the proposed method. Evaluation is detailed in section 4.

II. Background and Related Work

Unsupervised rank fusion algorithms can be divided into 3 categories: score-based, rank-based and voting algorithms. Fox and Shaw [3] introduced score based fusion (CombSUM, CombMAX, CombMNZ and other variants). Score based approaches use document score as the basis for scoring. The best performing approaches are combSUM, combMAX and combMNZ. For each document *i*, the score after fusion can be computed as:

$$\text{combSUM}(i) = \sum_{k=1}^{N(i)} S_k(i), \quad (1)$$

$$\text{combMAX}(i) = \max(S), \forall S \subset D_i, \quad (2)$$

$$\text{combMNZ}(i) = N(i) \times \text{combSUM}(i), \quad (3)$$

where $S_k(i)$ is the score of the *i* document on the result list (ranking) *k*. *N(i)* refers to the number of times a document appears on rankings (frequency). *N(i)* varies between 0 (the document *i* does not appear on any ranking) and the total number of rankings (document *i* appears on all rankings). *D(i)* are the rankings that contain document *i*. CombMNZ is the best performing score-based technique and a good example of the Chorus effect. The final score (sum of scores from all engines) is multiplied by *N(i)*. Thus, document score grows linearly with frequency. These approaches have been widely researched and evaluated for a long time [9], [10]. There are some disadvantages in score-based approaches: rankings must score documents. Score distribution is different across systems and normalization greatly influences performance [10]. This affects the skimming effect, as the score of the document may not necessarily reflect its rank, even after normalization. Consider this example with two engines with scores between [0,1]; in engine A, document D might be ranked 2 and have a score of 0.98 and in engine B, document D may also be ranked 2 but have a score of 0.8.

Rank based approaches are inspired by score based approaches, but use the (inverted) ranks as the scores. The inverse ranks reflect the skimming effect and avoid the problem of score normalization. Reciprocal Rank [4] and Reciprocal Rank Fusion [1] are the most widely known approaches. Rank based approaches tend to outperform score based approaches [1], [11] and learning to rank methods [1], [2]. Rank based approaches can also be deployed on use-cases where the score is not available. Rank based fusion methods consider the inverse of the rank of each document in each one of the individual lists as the score. Reciprocal Rank (RR) and Reciprocal Rank Fusion (RRF) are the techniques described on the literature. In contrast to the score-based techniques, these techniques do not require normalization and are more stable.

$$\text{RR}(i) = \sum_{k=1}^{N(i)} \frac{1}{R_k(i)}, \quad (4)$$

$$\text{RRF}(i) = \sum_{k=1}^{N(i)} \frac{1}{h + R_k(i)}, \text{with } h = 60. \quad (5)$$

where $R_k(i)$ is the rank of document *i* on the *k* rank.

Election algorithms are based on election theory and consider the individual ranks as votes. The most well known approaches are CondorFuse [12], based on Condorcet voting and Bordafuse [13]. As rank-based approaches, these algorithms only require the ranks (not scores) but are computationally more complex.

Learning to rank methods are supervised methods, requiring training data. Pairwise approaches like LambdaRank [14] focus on learning a classifier to rank document pairs. The goal is to create a final rank that minimizes misranked pairs. List-based approaches like ListNet [5] optimize the value of one of the evaluation metrics, using full ranks as input. In contrast, the proposed method has a much lower computational complexity and is fully unsupervised.

III. Inverse Square Rank

Our main objective is to improve rank-based data fusion by leveraging on the ideas behind the chorus and skimming effect, known techniques and prior work in score and rank-based techniques. We combined existing elements from those areas to improve retrieval performance. Inverse Square Rank (ISR) combines the inverse rank approach of RR and RRF (using the inverse of the rank as the score) with the document frequency component of combMNZ (results that appear on multiple lists have higher weight). The Inverse Square Rank fusion is defined as,

$$\text{ISR}(i) = N(i) \times \sum_{k=1}^{N(i)} \frac{1}{R_k(i)^2}, \quad (6)$$

where *N(i)* is the number of times a document appears on a results list (document frequency), and $R_k(i)$ is the rank of document *i* on the *k*th rank.

To visualize document scores assigned by ISR and other rank-based fusion algorithms, we mapped the score values for the top 50 ranked documents in Figure 1. The RRF scores decrease slowly with the rank position. A document ranked in position 50 will have less than half of the score of the document ranked first in a list. The RR function presents a faster decay; a document ranked at position 10 has a score that is 10 times smaller than a document ranked at the first position; a document ranked 50 gets an almost negligible score. In ISR, the curve slope is even greater. This is a fundamental characteristic to merge rank lists that have high precision. Search engines are very good on the top ranked results and differences in the ranking of documents towards the end of the result list start to lose significance. By boosting the top ranked documents score and multiplying by the frequency of the document across different result lists, we guarantee that documents that are highly ranked (higher probability of relevance to the query) and appear on multiple lists (to exclude documents from non-relevant engines or erratic documents) are ranked on top of the final fused result list.

The base form of the ISR weights the document frequency using the absolute number of matched lists. We observed that linear weighting over-emphasises documents present on multiple ranks and fails to penalize documents that appear in a single rank. In our initial experiments, penalizing documents that appear on a single rank, combined with logarithmic document frequency weighting, leads to a significant performance improvement. Inspired by BM25L [15] (where the logarithm was introduced to counteract increased score on long documents), we identified logarithmic normalization to provide

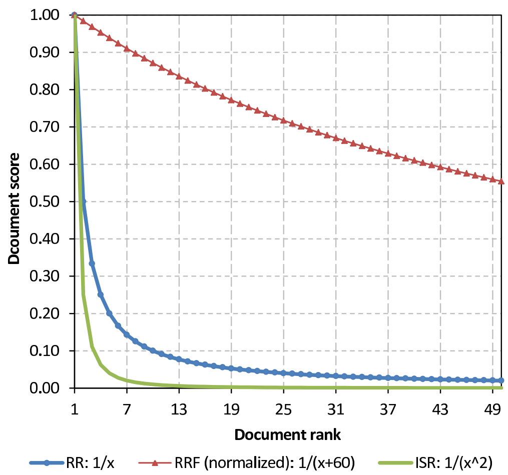

Fig. 1. Rank versus score using multiple functions.

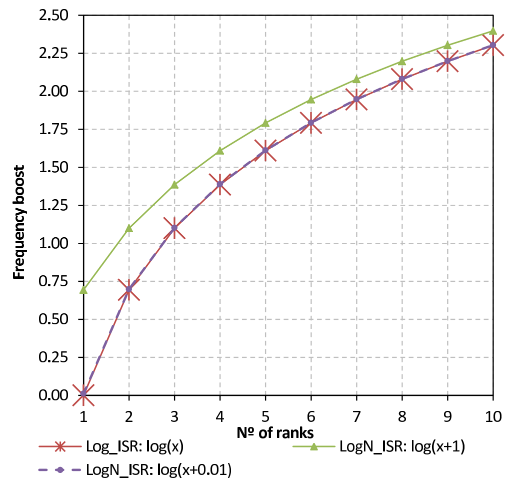

Fig. 2. Multiple document frequency weighting functions.

a good model for document frequency weighting. The final ISR techniques are log_ISR and normalized log_ISR (logN_ISR):

$$\text{log\_ISR}(i) = log(N(i)) \times \sum_{k=1}^{N(i)} \frac{1}{R_k(i)^2} \tag{7}$$

$$\text{logN\_ISR}(i) = log(N(i) + \sigma) \times \sum_{k=1}^{N(i)} \frac{1}{R_k(i)^2}. \tag{8}$$

Figure 2 represents the evolution of the frequency factor, using multiple normalization functions, for documents ranked from position 1 to 10. log_ISR gives zero score to documents that appear on a single ranked list. logN_ISR normalization factor guarantees that all documents have a $score > 0$.

We tested $\sigma \in [0, 1]$, and mapped important σ values. At $\sigma = 1$ the weight difference between low frequency documents and high frequency was too small. At $\sigma = 0.01$, the weight difference is closer to the desired behavior: single rank documents are given a very small but non-zero weight. For the remaining cases, logarithmic weight grows as expected. We set $\sigma = 0.01$ for the experiments on this paper.

Log_ISR gives zero score to results that appear on a single rank, being dependent on a secondary ranking function. On this paper, and in line with our experiments, we sorted ties using a deterministic shuffle. This helps filter results from single engines that are not relevant to the query, removing "biased noise". In addition to not requiring real document scores, ISR methods can handle an arbitrary number of ranked lists with an arbitrary number of documents per list. In some tasks, some ranked lists do not contain relevant results for some queries, which decreases global retrieval performance. A further performance boost could be achieved using supervised techniques to give more weight to ranks that have the best performance or that are more adequate for a certain query (dark horse effect). This could be applied in addition to our fusion technique, by multiplying the inverse rank score by the individual rank weight. This is out of scope for this paper, since we are only focusing on fully unsupervised approaches.

IV. Evaluation

The experiments are designed to assess the relative performance of our approach (ISR) to the existing data fusion approaches. We compared ISR to rank-based fusion (RR and RRF), score-based fusion (combSUM, combMAX, combMNZ) and voting-based fusion (CondorFuse) approaches. The ImageCLEF Medical Task 2013 dataset was used in our experiments.

A. *ImageCLEF Medical dataset.*

The dataset of ImageCLEF Medical [16] contains over 75,000 journal articles and over 300,000 images of biomedical open access literature from PubMed. Text data was preprocessed by removing stop-words and stemming. Standard tf-idf weighting and BM25 retrieval model was used. Image data was analysed and visual features were extracted: CEDD, Gabor Moments, FCTH and HSV marginal histograms. Search was implemented on an Euclidean space.

B. *Metrics.*

We chose standard IR metrics, including MAP and precision at certain cutoffs, and task specific metrics such as bpref which is used in the medical retrieval evaluation task. The evaluation was performed using the trec_eval implementation of the metrics and the relevance judgments provided in each dataset.

C. *Protocol.*

The protocol is the following: for each query, individual documents in all ranks are grouped by a unique id, resulting on a list containing the scores and ranks for that document on all ranks, Figure 3 (a). Final document scores are computed from the lists using the described approaches and the final rank is sorted by the resulting scores, Figure 3 (b). When documents obtain the same final score, the ordering on the final list is stable, meaning that running the same fusion algorithm with the same data will return the same results. Fused ranks are limited to 1000 documents.

Fig. 4. Visual query example (Osteoporosis x-ray images).

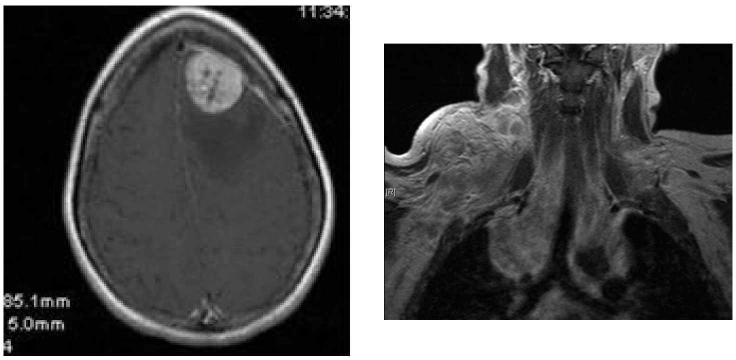

Fig. 5. Visual query example (Hodgkin's lymphoma, MRI images).

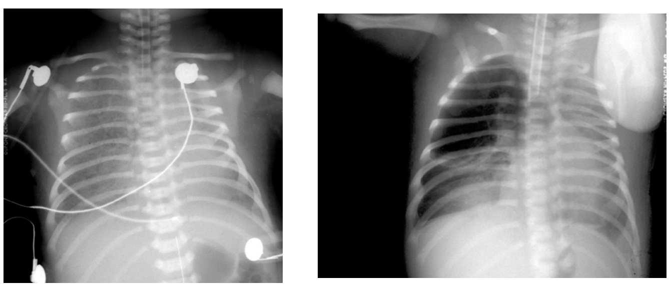

Fig. 6. Case-based query example. Visual examples and case description: "*A newborn with a lung lesion diagnoses on prenatal ultrasound. X-ray on the first day of life shows dense lungs bilaterally. X-ray on second day of life shows an air-containing cystic area occupying the right upper lung.*"

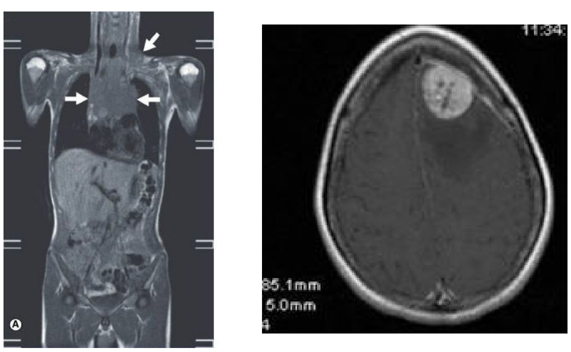

Fig. 7. Case-based query example. Visual examples and case description: "*A 36-year-old male with HIV, with a several week history of progressive dementia, apraxia, and visual deficits. T2-weight MRI shows a hyperintense lesion in the left frontoparietal region. CT scan shows a hypoattenuated lesion in the same location with a scalloped lateral margin.*"

D. Experiments

In this experiment, the objective is to combine the results from multimodal queries (text and images), using the previously described fusion approaches. The experiment consists of two tasks:

- In ad-hoc image retrieval task, the query consists of 1 to 7 sample images and a short text description of the medical diagnosis. The goal is to retrieve images that suit the query. There is a total of 35 queries (see examples on Figure 4 and 5).
- In case-based retrieval task, the query consists of a case description (with patient demographics, limited symptoms and test results including imaging exams). There is a total of 36 queries (see examples on Figure 6 and Figure 7), containing textual description and 2 to 6 images each. The goal is to retrieve relevant cases (in this case a PubMed article).

The fusion for both tasks is similar: the text and images are indexed and searched separately, resulting in two ranks per query (one for image search and another for text search).

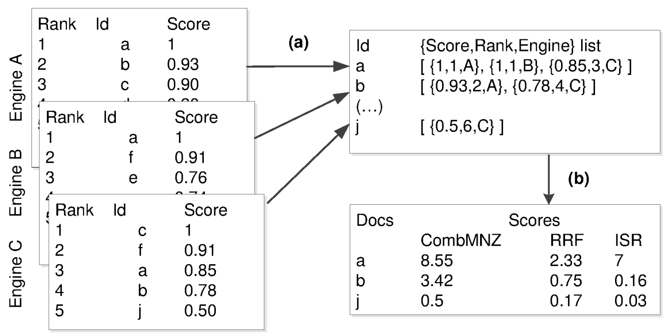

Fig. 3. Rank-based fusion example. Initially, the documents in the ranked lists are grouped by unique document identifier (id) (a). Individual document score is computed from the Score, Rank, Engine lists and the new grouped rank is sorted by new scores (b).

The two ranks are combined using the described approaches. Document unique identifiers (ids) are based on its Digital Object Identifier (DOI), meaning that each document has the same id across ranked lists.

TABLE I. IMAGECLEF MEDICAL CASE RETRIEVAL FUSION RESULTS.

Run	MAP	bpref	P@10	P@30
LogN_ISR	0.1599	0.1399	0.1657	**0.1286**
Log_ISR	0.0750	0.0854	0.1171	0.0790
ISR	**0.1608**	**0.1426**	**0.1800**	0.1257
RR	0.1583	0.1367	0.1771	0.1238
RRF	0.1505	0.1218	0.1600	0.1248
combMNZ	0.0804	0.0866	0.1429	0.0819
combSUM	0.0803	0.0857	0.1457	0.0790
combMAX	0.0292	0.0344	0.0457	0.0248
CondorFuse	0.0433	0.0732	0.0657	0.0448

1) Case-based retrieval.: Table I contains the results for the multimodal fusion experiment on the ImageCLEF case-based retrieval task. The rank-based approaches (RR, RRF, ISR, LogN_ISR) double the performance of the score-based approaches. RR obtains good performance on most metrics, but it loses on MAP to ISR methods. RR slightly outperforms RRF on most metrics, and on P@10 where RRF shows a significant loss. This is most likely due to the fact that in RRF, the document score decreases slower with rank than with RR or ISR, leading to less than optimal scoring in the top ten documents. LogN_ISR improves RRF marginally. ISR is the better method on most metrics. It attains the best MAP and P@10, being second only to LogN_ISR in terms of P@30. Score-based approaches (combSUM, combMNZ, combMAX) do not perform well on this task by any metric. The reason could be the instability of the scores and the distribution differences between the crossmedia ranks. CondorFuse performance was low as well. We reckon it might be due to having only two ranks for the voting scheme. Log_ISR performs in line with score-based approaches due to the implicit instability of this function when the number of ranks to combine is small (2 in this case). Only the results that occur on more than one list are ranked with expected precision. In this task, the frequency of the documents across lists is a key factor. ISR boosts results that appear on the two lists linearly, leading to the best performance.

2) Image retrieval.: Table II contains the results for the multimodality fusion for the ImageCLEF image retrieval task. The relative performance of the fusion algorithms is close to case-based retrieval. In this task, RRF slightly outperformed other approaches; we think because there are less documents in both ranks, reducing the importance of the document frequency weight. LogN_ISR is the second best method. ISR and RR come in third and fourth place respectively. Score based approaches and Log_ISR do not perform well again, being outperformed by rank-based approaches.

TABLE II. IMAGECLEF MEDICAL IMAGE RETRIEVAL FUSION RESULTS.

Run	MAP	bpref	P@10	P@30
LogN_ISR	0.1482	0.1529	0.2143	**0.1543**
Log_ISR	0.0475	0.0924	0.1400	0.0648
ISR	0.1458	0.1505	0.2057	0.1476
RR	0.1450	0.1493	0.2086	0.1457
RRF	**0.1508**	**0.1557**	**0.2171**	**0.1543**
combMMZ	0.0489	0.0739	0.1457	0.0752
combSUM	0.0480	0.0730	0.1343	0.0733
combMAX	0.0246	0.0317	0.0629	0.0400
CondorFuse	0.0182	0.0493	0.0314	0.0438

The precision-recall curves for case and image retrieval are presented in Figure 8 and 9, respectively. The precision-recall curves follow the global results closely. ISR, logN_ISR, RRF and RR have roughly the same curve. The remaining algorithms' curves are located below. On all algorithms, precision drops sharply at 10% recall, meaning that the rank fusion methods are better at ranking the top positions.

V. CONCLUSIONS AND DISCUSSION

In this paper, we proposed the Inverse Square Rank data-fusion technique. It is based on the quadratic decay of a document score and on the logarithmic document-frequency normalization. We evaluated its performance using standard IR datasets and compared it with the current state-of-art algorithms.

LogN_ISR is the most balanced solution of the tested use-cases, outperforming one of the best existing algorithms (RRF) on most metrics in our experiments. Log_ISR performed poorly in this dataset. The most likely cause is the random document sorting for documents that appear on a single ranking. Highly ranked documents and document-frequencies across different lists are the most important qualities of documents in rank-based fusion. The ISR algorithm shares the simplicity of RRF with improved retrieval performance through logarithmic document frequency normalization with Log_ISR and LogN_ISR.

Acknowledgements. This work has been partially funded by the projects PTDC/EIA-EIA/111518/2009 and UTA-Est/MAI/0010/2009 funded by the Portuguese National Foundation for Science and Technology (FCT).

REFERENCES

[1] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *SIGIR '09*. New York, NY, USA: ACM, 2009, pp. 758–759. [Online]. Available: http://doi.acm.org/10.1145/1571941.1572114

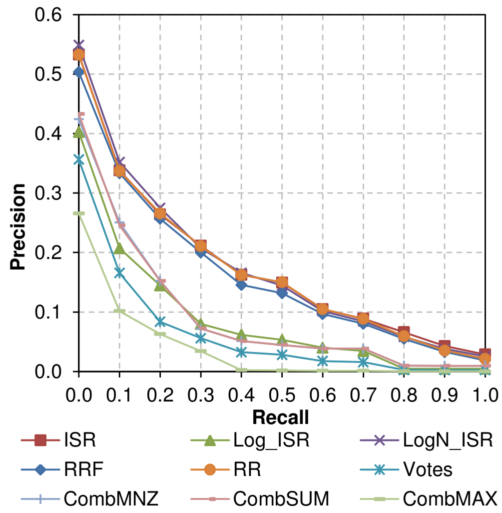

Fig. 8. ImageCLEF 2013 medical case retrieval precision-recall curve

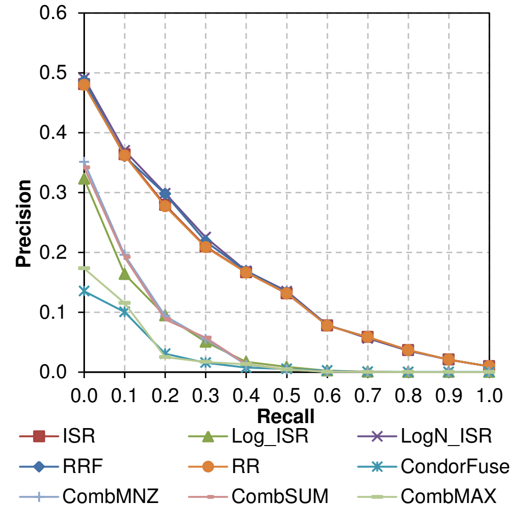

Fig. 9. ImageCLEF 2013 medical image retrieval precision-recall curve

[2] C. Macdonald and I. Ounis, "Voting for candidates: adapting data fusion techniques for an expert search task," in *Proceedings of the 15th ACM international conference on Information and knowledge management*, ser. CIKM '06. New York, NY, USA: ACM, 2006, pp. 387–396. [Online]. Available: http://doi.acm.org/10.1145/1183614.1183671

[3] J. A. Shaw and E. A. Fox, "Combination of multiple searches," in *The Second Text REtrieval Conference (TREC-2*, 1994, pp. 243–252.

[4] M. Zhang, R. Song, C. Lin, S. Ma, Z. Jiang, Y. Jin, Y. Liu, and L. Zhao, "Expansion-based technologies in finding relevant and new information: Thu trec2002 novelty track experiments," in *the Proceedings of the Eleventh Text Retrieval Conference (TREC*, 2002, pp. 586–590.

[5] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 129–136. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273513

[6] J. Xu and H. Li, "Adarank: a boosting algorithm for information retrieval," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '07. New York, NY, USA: ACM, 2007, pp. 391–398. [Online]. Available: http://doi.acm.org/10.1145/1277741.1277809

[7] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, Dec. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=945365.964285

[8] C. C. Vogt and G. W. Cottrell, "Fusion via a linear combination of scores," *Inf. Retr.*, vol. 1, no. 3, pp. 151–173, Oct. 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1009980820262

[9] J. H. Lee, "Analyses of multiple evidence combination," in *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '97. New York, NY, USA: ACM, 1997, pp. 267–276. [Online]. Available: http://doi.acm.org/10.1145/258525.258587

[10] M. Montague and J. A. Aslam, "Relevance score normalization for metasearch," in *Proceedings of the tenth international conference on Information and knowledge management*, ser. CIKM '01. New York, NY, USA: ACM, 2001, pp. 427–433. [Online]. Available: http://doi.acm.org/10.1145/502585.502657

[11] D. Frank Hsu and I. Taksa, "Comparing rank and score combination methods for data fusion in information retrieval," *Inf. Retr.*, vol. 8, no. 3, pp. 449–480, May 2005. [Online]. Available: http://dx.doi.org/10.1007/s10791-005-6994-4

[12] M. Montague and J. A. Aslam, "Condorcet fusion for improved retrieval," in *Proceedings of the eleventh international conference on Information and knowledge management*, ser. CIKM '02. New York, NY, USA: ACM, 2002, pp. 538–548. [Online]. Available: http://doi.acm.org/10.1145/584792.584881

[13] J. A. Aslam and M. Montague, "Models for metasearch," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '01. New York, NY, USA: ACM, 2001, pp. 276–284. [Online]. Available: http://doi.acm.org/10.1145/383952.384007

[14] C. J. C. Burges, R. Ragno, and Q. V. Le, "Learning to Rank with Nonsmooth Cost Functions," in *NIPS*, B. Schölkopf, J. C. Platt, T. Hoffman, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2006, pp. 193–200. [Online]. Available: http://dblp.uni-trier.de/rec/bibtex/conf/nips/BurgesRL06

[15] Y. Lv and C. Zhai, "When documents are very long, bm25 fails!" in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 1103–1104.

[16] A. Garcia Seco de Herrera, J. Kalpathy-Cramer, D. Demner Fushman, S. Antani, and H. Müller, "Overview of the imageclef 2013 medical tasks." in *Working notes of CLEF 2013*, 2013.