Combining Strategies for XML Retrieval

Ning Gao[1], Zhi-Hong Deng[1,2,⋆], Jia-Jian Jiang[1], Sheng-Long Lv[1], and Hang Yu[1]

[1] Key Laboratory of Machine Perception (Ministry of Education),
School of Electronic Engineering and Computer Science, Peking University
[2] The State Key Lab of Computer Science, Institute of Software,
Chinese Academy of Sciences, Beijing 100190, China
{ninggao,zhdeng,jiangjiajian}@pku.edu.cn,
{davidfracs,pkucthh}@gmail.com

Abstract. This paper describes Peking University's approaches to the Ad Hoc, Data Centric and Relevance Feedback track. In Ad Hoc track, results for four tasks were submitted, Efficiency, Restricted Focused, Relevance In Context and Restricted Relevance In Context. To evaluate the relevance between documents and a given query, multiple strategies, such as Two-Step retrieval, MAXLCA query results, BM25, distribution measurements and learn-to-optimize method are combined to form a more effective search engine. In Data Centric track, to gain a set of closely related nodes that are collectively relevant to a given keyword query, we promote three factors, correlation, explicitnesses and distinctiveness. In Relevance Feedback track, to obtain useful information from feedbacks, our implementation employs two techniques, a revised Rocchio algorithm and criterion weight adjustment.

Keywords: INEX, Ad Hoc, Data Centric, Relevance Feedback.

1 Introduction

INEX Ad Hoc Track [1] aims at evaluating performance in retrieving relevant results (e.g. XML elements or documents) to a certain query. In Ad Hoc 2010, four different tasks are addressed: (1) Efficiency task requires a thorough run to estimate the relevance of documents components. (2)Relevant in Context task requires a ranked list of non-overlap XML elements or passages grouped by their corresponding parent articles. (3) Restricted Focused task limits results (elements or passages) ranked in relevance order up to a maximal length of 1,000 characters per topic. (4) Restricted Relevant in Context tasks requires a ranked list of elements or passages. And for each element a ranked list result covers its relevant material, at most 500 characters for each result document.

Initially we only consider the effectiveness of retrieval, regardless of the efficiency and restricted length. Five different querying strategies, BM25 [2], Two-Step retrieval, Maximal Lowest Common Ancestor (MAXLCA)[3] query results, distribution measurements and learn-to-optimize method, are combined to form a more efficient search engine. Detailed definitions of these technologies is introduced in section 2. The thorough retrieval results are submitted to efficiency task. Furthermore, the results for other

[⋆] The corresponding author.

S. Geva et al. (Eds.): INEX 2010, LNCS 6932, pp. 319–331, 2011.

three tasks are all obtained based on thorough run task. In Restricted Focused task, each topic limits the answers up to a maximal length of 1,000 characters. Hence, we only return the result elements with top relevance-length ratio, in which relevance score for each results are computed by the aforementioned combined strategies module. For Relevance in Context task, the process scans the thorough runs and integrates the elements belonging to the same document. The orders for these integrated elements in ranked list are determined by the elements with maximal relevance score in each set. To obtain the results of Restricted Relevance in Context task, each result in the Relevance in Context is pruned to maximal 500 characters. Similar to the Restricted Focused task, only passages with top relevance-length ratio are retrieved.

Data Centric track is to provide a common forum for researchers or users to compare different retrieval techniques on data-centric XML documents, whose structure is plentiful and carries important information about objects and their relationships. In order to generate a good snippet [4] of an XML document, it is critical to pick up the most descriptive or representative attributes together with their values. In this paper, we present three main principles to judge the importance of attributes:(1) whether the attribute is distinguishable or not; (2)whether the attribute is explicit or implicit; (3)whether the attribute describes the entity directly or indirectly.

Relevance feedback[1] is a new track in INEX 2010. IR system with relevance feedback permits interactivities between the users and the system. Users provide relevance (irrelevance) information of search result to IR system, which is utilized by IR system to return more effective results. Relevance feedback track in INEX2010 simulates a single user searching for a particular query in an IR system that supports relevance feedback. The user highlights relevant passages of text and provides this feedback to the IR system. The IR system re-ranks the remainder of the unseen result list to provide more relevant results to the user. The Relevance Feedback track mainly focuses on the improvement of search result before and after implementing relevance feedback. Consequently, our team pay more attention to acquiring more information through feedback rather than optimizing results as what we did in Ad hoc track.

In section 2, we explicitly introduce the five strategies used in Ad Hoc track and reveal the corresponding evaluation results. Section 3 describes our work on Data Centric track. In section 4, the methods applied for Relevance Feedback track are presented.

2 Ad Hoc Track

Figure 1 describes the framework of our search engine. The Inverted Index of the Data Collection is initially processed in the background. Afterwards, when a user submits a Query to the Interface, the search engine first retrieves the relevant documents. Based on the definition of MAXimal Lowest Common Ancestor (MAXLCA) and All Common Ancestor (ACA) query results, the relevant elements are extracted from the relevant documents. Furthermore, we use the ranking model BM25 to rank these extracted disordered element results, the output of the processing on a Ranked List. To further improve the effect of the ranking module, the top several results in the Ranked List are re-ranked by Distribution Measurements. In Distribution Measurements, there are four criterions based on the distribution of keywords that are taken into consideration, explicitly introduced in section 4. The weights of these four criterions in the re-ranking function

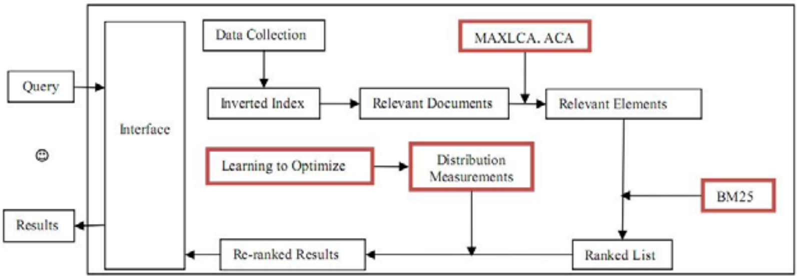

Fig. 1. System Framework

are trained by a Learning to Optimize method. Finally, the Re-ranked Results are returned to the user as searching Results. In the search engine, five different strategies are used, Two-Step search, MAXLCA query results, BM25, Distribution Measurements and Learn to Optimize method.

- **Two-Step Retrieval:** Different from HTML, the retrievable unit for the INEX focused task is XML elements rather than the whole document text. Therefore, the core idea of Two-Step retrieval is splitting the searching process into two steps. The first level starts from the traditional article retrieval. Then taking the top returned relevant articles as querying database, the second layer further processes extracting the relevant elements. Finally, the extracted elements are ranked and returned in a result list form. In INEX 2009, one of the most effective search engines proposed by Queensland used Two-Step as retrieval strategy[19].
- **BM25:** Based on various research and comparative experiments, BM25 is confirmed to be an effective ranking method. It takes both text and structure information into consideration. Additionally, evaluation results of Ad Hoc Track show that BM25 performs better than some other frequently cited ranking models, such as tf*idf [5] etc. Motivated by BM25's excellent performance, we implant it into the search engine as a basic ranking method.

In the remainder of the section, we apply other three technologies in the search engine. MAXLCA and ACA defines which elements are relevant and to be returned as results. Distribution measurements are re-ranking criterions used to evaluate the relevance between elements and queries according to the distribution of the keyword matches. Learn-to-optimize method is devised to tune the weights of different ranking methods in the final decision.

2.1 Maximal Lowest Common Ancestor (MAXLCA)

Due to the fact that the returned results of XML retrieval are elements, an adaptive XML search engine should define which elements in the XML tree are relevant and to be retrieved. Several approaches have been proposed for identifying relevant results, such as XRANK [6], SLCA [7] and XSeek [8] and so on. In this paper, our definition of query results are called MAXLCA. Furthermore, we compare it with another widely used definition of query results, naming All Common Ancestor (ACA).

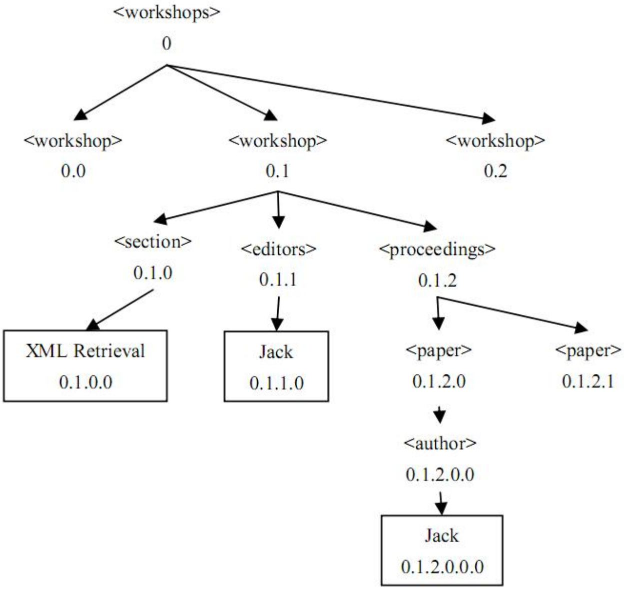

Fig. 2. Sample XML Tree

- **ACA:** In an XML tree, the nodes containing all matches of keywords are returned as relevant results. For example, there is a submitted query {XML retrieval, Jack} and an XML tree presented as figure 2, in which the matches of keywords have been marked. Node 0 and 0.1 contains all the three matches in the tree, so that these two nodes should be returned.
- **MAXLCA:** In an XML tree, the lowest node that contains all matches of keywords is defined as a query result. For example, within the nodes containing all matches of keywords in XML tree, 0.1 is the lowest one. Therefore, node 0.1 is the unique query result.

2.2 Distribution Measurements

By observing a large amount of over 2000 query-result pairs, we discover that the distribution of the keyword matches in results plays a crucial role in picturing the theme of the passage. Moreover, four detailed statistical characteristics based on distribution are considered, which present advantage capability on distinguishing relevant and irrelevant passages.

- **Distance Among Keywords (DAK).** Zhao etc. considers the proximity of query terms in [20]. Here in this paper the minimum distance among the keywords is calculated. The closer the matches of keywords, the more relevant an element is.
- **Distance Among Keyword Classes (DAKC).** Xu and Croft discuss term clustering in [21]. In this paper, the matches of a certain keyword in passages are firstly clustered into several subsets. The closer the keywords subsets are, the more relevant this passage is.

- **Degree of Integration Among Keywords (DIAK).** The passage with higher degree of integration is considered as more concentrating on one certain theme and should be given a higher priority in the returned list.
- **Quantity Variance of Keywords (QVK).** The passages whose numbers of different keywords vary significantly should be penalized.

2.3 Learn to Optimize the Parameters

Nerual Network method in machine learning is introduced to tune the weights of the four features in distribution measurements. The Wiki English collection, queries and assessments of INEX 2009 Ad Hoc track are used as training samples.

In training, there is a set of query $Q = \{q_1, q_2, ..., q_m\}$ extracted from the INEX 2009 Ad Hoc track. Each query q_i is associated with a list of candidate elements $E_i = (e_i^1, e_i^2, ..., e_i^{n(i)})$, where e_i^j denotes the the j-th candidate element to query q_i and $n(i)$ is the size of E_i. The candidate elements are defined as MAXLCA or ACA elements. Moreover, each candidate elements list E_i is associated with a ground-truth list $G_i = (g_i^1, g_i^2, ..., g_i^{n(i)})$, indicating the relevance score of each elements in E_i. Given that the wiki collection only contains information of whether or not the passages in a document is relevant, the F-measure [22] is applied to evaluate the ground truth score. Given a query q_i, the ground-truth score of the j-th candidate element is defined as follows:

$$precision = \frac{relevant + irrelevant}{relevant} \tag{1}$$

$$recall = \frac{relevant}{REL} \tag{2}$$

$$g_i^j = \frac{(1 + 0.1^2) \cdot precision \cdot recall}{0.1^2 \cdot precision + recall} \tag{3}$$

In the formula, *relevant* is the length of relevant contents highlighted by user in e, while *irrelevant* stands for the length of irrelevant parts. *REL* indicates the total length of relevant contents in the data collection. The general bias parameter is set as 0.1, denoting that the weight of precision is ten times as much as recall.

Furthermore, for each query q_i, we use the distribution criterions defined in section 2.2 to get the predicted relevant scores of each candidate element, recorded in $R_i = (r_i^1 r_i^2, ..., r_i^{n(i)})$. In formula (4), S_{DAK}, S_{DAKC}, S_{DIAK} and S_{QVK} are the predicted scores for element j according to distance among keywords, distance among keyword classes, degree of integration among keywords and quantity variance of keywords respectively.

$$r_i^j = \alpha S_{DAK} + \beta S_{DAKC} + \gamma S_{DIAK} + \delta S_{QVK} \tag{4}$$

Then each ground truth score list G_i and predicted score list R_i form a "instance". The loss function L is defined as the Euclidean distance between standard results lists D_i and search results lists R_i. In each training epoch, the four criterions were used to compute the predicted score R_i. Then the learning module replaced the current weights with the new weights tuned according to the derivative of the loss between G_i and R_i. Finally the process stops either when reaching the limit cycle index or the parameters do not change. Precise descriptions see [9].

2.4 Comparison Results

According to (1) two-step strategy or simple element retrieval; (2) ACA results or MAXLCA results; (3) BM25, Distribution or BM25+Distribution, there should be 12 kinds of combination methods altogether. However, due to some previous experiments, we only submit 5 different combinations which are predicted as effective to Ad Hoc track, illustrated in table 1.

Table 1. Results Submitted to Ad Hoc Track

	MAXLCA	ACA	Two-Step	BM25	Distribution
AcaBM25		✓		✓	
MaxBM25	✓			✓	
RefMaxDis	✓		✓		✓
RefMaxBM25	✓		✓	✓	
RefMaxBM25Dis	✓		✓	✓	✓

Figure 3,4,5 illustrate the evaluation results of Efficiency task, Relevance In Context task and Restricted Relevance In Context task respectively under measure as focused retrieval. For Restricted Focused task, since we only submitted the RefMaxBM25Dis results, there is no useful and convincing comparison results that can be shown. However, as can be concluded from the other three tasks:

- Two-Step search performs better than simple element search. According to tradition ranking methods, such as BM25 and tf*idf, the elements with high (tf / passage length) ratio are marked as high relevance, emphasizing on small fragments even if they are extracted from irrelevant documents. Two-Step strategy eliminates such bias, since the candidate elements are all extracted from documents predicted to be relevant.
- MAXLCA performs better on wiki collection than ACA. Though according to our experiments on the data collection of INEX2010 and INEX2009, the MAXLCA results perform better than ACA results, we still support the definition of ACA for its apparently high flexibility. On the other hand, the definition of MAXLCA is only suitable for short documents, such as web pages.
- Rather than completely abandoning BM25, distribution measurement is suitable for improving the performance and modifying the drawbacks of it. Our approach originally aims at modifying the disadvantage parts of BM25, since it has been proved effective by many searching systems in INEX. The distribution measurement is a re-ranking method, where each standards only focuses on one single point. Accordingly, we use a learning method to learn the optimal weights of these standards for a certain data collection and only in this way, the final re-ranking method are actually determined by the data collection.
- The method using Two-Step as retrieving strategy, MAXLCA as query results, BM25 and distribution measurement as ranking functions shows the best performance.

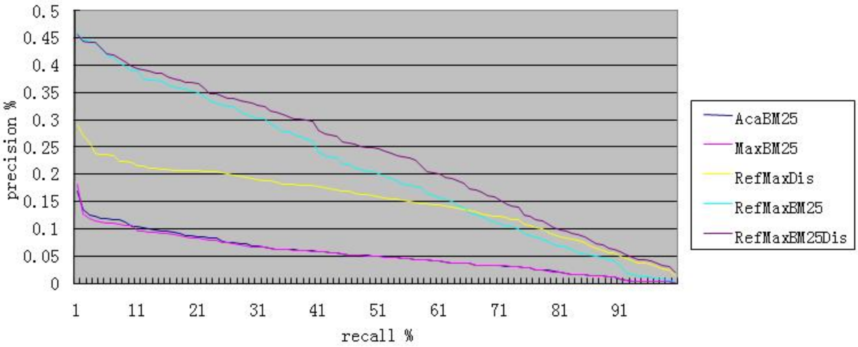

Fig. 3. Evaluation Results of Efficiency Task

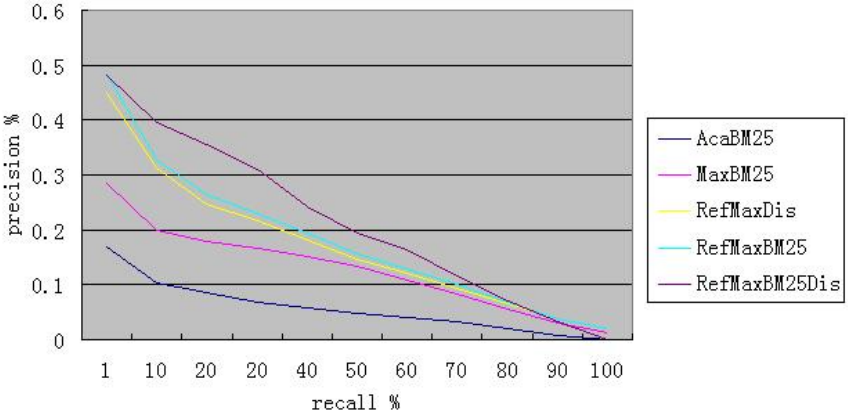

Fig. 4. Evaluation Results of Relevance in Context Task

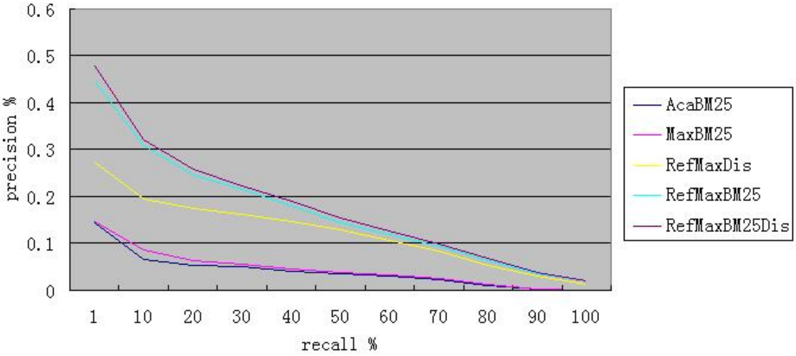

Fig. 5. Evaluation Results of Restricted Relevance in Context Task

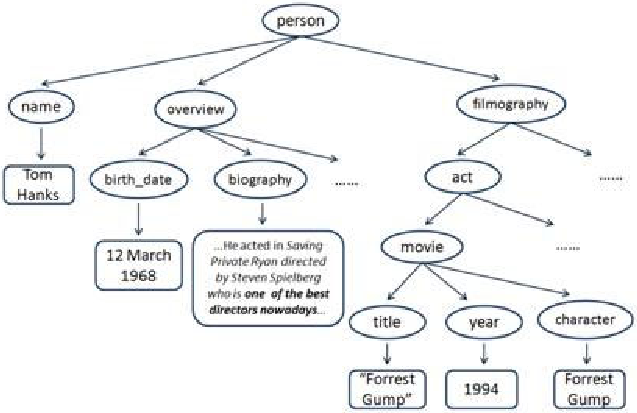

Fig. 6. IMDB data segment

3 Data Centric Track

In XML keyword search, there are quite numbers of results returned, only by the snippet of each result can the users judge whether it meets the search intention. Therefore, it is important to generate a good snippet for each result. [8][10] pointed that a good XML result snippet should be a self-contained information unit of a bounded size that effectively summarizes the query result and differentiates itself from the others.

3.1 Related Work

[10] [11] pointed out that a good XML result snippet should be a self-contained information unit of a bounded size that effectively summarizes the query result and differentiates itself from the others. Meanwhile the authors have also accomplished a snippet generation system called eXtract [4]. The size limitation of a snippet requires the search engine to extract the most important information (information here refers to attributes of an entity) from the result document, thus the main problem is to define the importance of attributes.

3.2 Extracting the Most Representative Attributes

In this paper, we use a semantic model MRepA (Most Representative Attribute)[12] to evaluate the importance of an attribute to its corresponding entity. Further, three main principles are proposed to judge the importance of attributes, presented as, (1)whether the attribute is distinguishable or not; (2)whether the attribute is explicit or implicit; (3)whether the attribute describes the entity directly or indirectly.

3.2.1 Correlation between Entities and Attributes

To evaluate whether an attribute describes an entity directly, we analyze the position between the attribute and the entity in an XML document tree. In this section we define the entity-attribute path and use the number of entities on the path to measure the *correlation* between an attribute and its corresponding entity.

Definition 1. *An entity-attribute path is a set of nodes which are on the path from an entity to its attribute(including the entity and the attribute).*

Definition 2. *We define the correlation between entity e and attribute a R(e, a) as follows*

$$R(e, a) = k^{length(e,a)} \cdot \prod_{i=1}^{n} \frac{1}{m_i} \tag{5}$$

Where $n = length(e, a)$ refers to the number of entities between entity e and attribute a, and m_i refers to the number of the entities of i-th category in the path. k is a parameter set less than 1.

For example, in figure 4, suppose that there are 10 movie nodes, the entity-attribute path from node person to title *Forrest Gump* is {person, filmography, act, movie, title}. There are two entities *person* and *movie* on the path. The number of person is 1, while the number of the movie is 10.

3.2.2 Explicitnesses of Attributes

In XML keyword search, the length of the value of an attribute is usually associated with the explicitness of the attribute. Long text tends to be more descriptive but less explicit, while short text tends to be more explicit and clear. Thus we use the length (or the number of words) of the text to judge the explicitness of an attribute roughly.

Definition 3. *We judge the explicitness of an attribute by the complicacy (length) of its value, and we denote the explicitness of attribute a as E(a).*

3.2.3 Distinctiveness of Attributes

A distinguishable attribute should match the following two conditions, (1) the attribute appears in all of the entities; (2) the values of the attribute are different in different entities. Due to the possibility that two different person share the same name, in this section we promote the formula to calculate how much an attribute meets the demands.

Definition 4. *We use distinctiveness of attributes to evaluate the distinguish ability of the attributes.*

$$W_a = exp(p_a) \cdot H(a) \tag{6}$$

$$H(a) = -\sum_{i=1}^{n} p(a_i) \cdot log[p(a_i)] \tag{7}$$

In the above formulas, W_a is the distinctiveness of attribute a. p_a refers to the percentage of the correlative entities where attribute *a* appears. H(a) is the entropy of attribute *a*, which estimates the variety intensity of attribute *a*.

3.2.4 MRepA Model

In MRepA model, we take the above three factors into consideration. Given a keyword query, we firstly return a set of entities as results, and then pick up the top-k most important attributes of each entity into the snippet.

Definition 5. *The importance of an attribute a to an entity e S(e,a) is defined as follows*

$$S(e,a) = W_a \times E(a)^{R(e,a)} \tag{8}$$

3.2.5 Comparison Results

In Data Centric track, there are three assessments, TREC MAP metric, INEX thorough retrieval MAiP metric and INEX Relevant-in-Context MAgP T2I(300). In MAP metric, our results perform the best, using the description and narrative of the topics as extra information. However, poor responses are got under MAiP and MAgP metric.

Table 2. Results Submitted to Data Centric Track

	MAP	MAgP	MAiP
MRepA	0.5046	NA	0.0157

4 Relevance Feedback Track

In relevance feedback track, we employs two techniques, a revised Rocchio algorithm and criterion weight adjustment. In section 4.1, we briefly introduce Rocchio algorithm [13]. In section 4.2 the revised algorithm is proposed. We discuss the adjustment of the criterion weights in section 4.3.

4.1 Rocchio Algorithm

Rocchio algorithm operates on vector space model, in which a document is represented by a vector of n weights such as $d = (t_1, t_2, t_3, t_4, ..., t_n)$. Where n is the number of unique terms in document collections and t_i is the weight of the i-th term in document d. The keyword query is also presented as a vector.

The general idea of Rocchio algorithm is that the initial query may not express the purpose of a IR system user completely and effectively. Rocchio algorithm's goal is to define an optimal query that maximize the difference between the average vector of the relevant documents and the average vector of the irrelevant documents. To achieve this, Rocchio algorithm adds new query terms and re-weight query terms in the query vector, making it more discriminative in choosing relevant documents from documents collection. The following formula shows the basic Rocchio algorithm

$$Q_t = \alpha Q_0 + \beta \frac{1}{n_1} \sum_{i=0}^{n_1} R_i - \gamma \frac{1}{n_2} \sum_{i=0}^{n_2} S_i \tag{9}$$

where Q_0 is the initial query vector and Q_t is the revised query vector, n_1 is the number of relevant documents and n_2 is the number of irrelevant documents, $R_(i)$ is the vector of a relevant document, S_i is a irrelevant document vector, and α, β, γ control the influence of each part.

After modification, the terms only appear in relevant document get a high positive weight and those only in irrelevant documents get a high negative weight, while the terms get relatively low weight if they appear in both relevant and irrelevant documents and have less discriminative power.

Some researchers have modified and extended the formula such as assigning different weight to original query terms[14] and added query terms or make constraints of number of documents used in the modification[15]. There are some other feedback methods based on probabilistic model and language model. The feedback method in classical probabilistic models is to select expanded terms primarily based on Robertson and Sparck-Jones Weight[16]. In language model, Zhai et al[17] estimate a model for all relevant documents together and expand original query model with it. There are also many other methods directly estimating a relevance model with relevance document[18].

4.2 Revised Rocchio Algorithm

Due to the fact that relevant information about each part of a relevant document is accessible in INEX2010 we divided a document into several paragraphs(we use "$< p >$" and "$< /p >$" to identify paragraphs) and represent each paragraph as a vector in our implementation. We treat a paragraph as a document in the searching process and give each paragraph a score. We also assign a weight to each paragraph according to its length. The score of a document is the weighted sum of its paragraphs' scores. In our implementation, we define query expansion formula as

$$Q_t = Q_0 + \frac{1}{n}\sum_{i=0}^{n} P_i \tag{10}$$

Where P_i donates the vector of term weights calculated from a paragraph. For term t_j in P_i, we define its weight as follows

$$w_j = \begin{cases} score_{P_i} & \text{if } P_i \text{ is a relevant paragraph in relevant document} \\ 0 - score_{P_i - t_j} & \text{if } score_{P_i} \text{ is a paragraph in irrelevant document} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Where $score_{P_i}$ denotes the score of paragraph P_i , and $score_{P_i - t_j}$ denotes the score of paragraph after removing all t_j from it.

Here we use an example to illustrate why we compute P_i like this. In INEX2009, the first topic of Ad hoc track is *Noble Prize*. A user queries this for preparing a presentation about the *Nobel Prize*. Therefore, he or she wants to collect information about *Nobel Prize* as much as possible. Assume that there is a document with a simple paragraph as a section title, *Ig Nobel Prize*. Apparently, the paragraph is not relevant because the *Ig Nobel Prize* is an American parody of the Nobel Prizes organized by the scientific humor magazine Annals of Improbable Research. However, the score of this paragraph is relatively high because it only contains three words and two of them are keywords. Intuitively, we can figure out that the term *Ig* is a bad word for this topic since it turns a high-score paragraph to a irrelevant one. In addition, term *Nobel* or *Prize* has no contribution to the irrelevance of this paragraph. However, if we use the formula in section 4.1, no difference between *Ig* and *Nobel* is reflected in the values of R_i or S_i. While in the revised model, the weights of *Ig* and *Nobel* are significantly different. In the revised model, we focus on the contribution of a term to relevance or irrelevance of the paragraph it belongs to.

4.3 Criterion Weight Adjustment

To calculate score of a paragraph, we make three criterions, the frequency entropy, the mixing entropy and the weighted term distance between paragraph vector and query vector. The frequency entropy scales difference of terms' appearance frequency. It assigns a high score if all keywords appear the same number of times in a paragraph. The mixing entropy scales the alternation of keywords. It assigns low score to a paragraph if it talks about one of the keyword at beginning while talks about another keyword at the end without a mixture of them. Each criterion makes contrition to the final score of a paragraph.

However, it is hard decide which criterion is of greater importance to a specific topic. So we try to get this information from the feedback data. In the searching process, we keep the score history of every criterion and every keyword. When updating the criterion weights, the discriminative power of each criterion and each keyword are computed. The discriminative power is computed as follows

$$DP = \frac{(\mu_r - \mu_{ir})^2}{d_r^2 - d_{ir}^2} \qquad (12)$$

Where μ_r is the mean contribution of this criterion or keyword to relevant paragraphs and μ_{ir} is the mean contribution of this criterion or keyword to irrelevant paragraphs. d_r is the standard deviation of contribution of this criterion or keyword to relevant paragraphs and d_{ir} is the standard deviation of contribution of this criterion or keyword to relevant paragraphs. High DP value means strong discriminative power in current topic, so we raise its weight to let it make bigger contribution to scoring paragraph. While low DP value indicates a criterion of keyword that is not suitable for the current topic.

For example, in our experiment, in the topic *Nobel Prize*, these two keywords are assigned the same criterion weight 0.5. However after all the document are returned, the criterion weight of *Nobel* is raised to 0.89 but the *Prize* is only 0.11. This is understandable. *Prize* is relatively a more widely used word because there are a lot of prizes such as Fields Medal Prize, Turing Prize.

Acknowledgments. This work was supported by the National High-Tech Research and Development Plan of China under Grant No.2009AA01Z136.

References

1. http://www.inex.otago.ac.nz/
2. Carmel, D., Maarek, Y.S., Mandelbrod, M., et al.: Searching XML documents via XML fragments. In: SIGIR 2003, pp. 151–158 (2003)

3. Gao, N., Deng, Z.H., Jiang, J.J., Xiang, Y.Q., Yu, H.: MAXLCA A Semantic XML Search Model Using Keywords. Technical Report
4. Huang, Y., Liu, Z., Chen, Y.: eXtract: A Snippet Generation System for XML Search. In: VLDB 2008, pp. 1392–1395 (2008)
5. Theobald, M., Schenkel, R., Wiekum, G.: An Efficient and Versatile Query Engine for TopX Search. In: VLDB 2005, pp. 625–636 (2005)
6. Guo, L., Shao, F., Botev, C., Shanmugasundaram, J.: XRANK: Ranked Keyword Search over XML Documents. In: SIGMOD 2003, pp. 16–27 (2003)
7. Xu, Y., Papakonstantinou, Y.: Efficient Keyword Search for Smallest LCAs in XML Databases. In: SIGMOD 2005, pp. 537–538 (2005)
8. Liu, Z., Chen, Y.: Identifying Meaningful Return Information for XML Keyword Search. In: SIGMOD 2007, pp. 329–340 (2007)
9. Gao, N., Deng, Z.H., Yu, H., Jiang, J.J.: ListOPT: A Learning to Optimize Method for XML Ranking. In: PAKDD 2010 (2010)
10. Liu, Z., Chen, Y.: Identifying Meaningful Return Information for XML Keyword Search. In: SIGMOD 2007, pp. 329–340 (2007)
11. Huang, Y., Liu, Z.Y., Chen, Y.: eXtract: A Snippet Generation System for XML Search. In: VLDB 2008, pp. 1392–1395 (2008)
12. Jiang, J., Deng, Z.H., Gao, N., Lv, S.L., Yu, H.: MRepA: Extracting the Most Representative Attributes in XML Keyword Search. Technical Report
13. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. The Knowledge Engineering Review 18(2), 95–145 (2003)
14. Ide, E.: New experiments in relevance feedback. In: Salton, G. (ed.) The SMART Retrieval System Experiments in Automatic Document Processing, ch. 16, pp. 337–354 (1971)
15. Ide, E., Salton, G.: Interactive search strategies and dynamic file organization in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System - Experiments in Automatic Document Processing, ch.18, pp. 373–393 (1971)
16. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. Journal of the American Society of Information Science 27(3), 129–146 (1976)
17. Zhai, C., Lafferty, J.D.: Model-basedfeedback in the language modeling approach toinformation retrieval. In: CIKM 2001, pp. 403–410 (2001)
18. Lavrenko, V., Bruce Croft, W.: Relevance-basedlanguage models. In: SIGIR 2001, pp. 120–127 (2001)
19. Geva, S., Kamps, J., Lethonen, M., Schenkel, R., Thom, J.A., Trotman, A.: Overview of the INEX 2009 ad hoc track. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2009. LNCS, vol. 6203, pp. 4–25. Springer, Heidelberg (2010)
20. Zhao, J., Yun, Y.: A proximity language model for information retrieval. In: SIGIR 2009, pp. 291–298 (2009)
21. Xu, J., Croft, W.B.: Improving the effectiveness of information retrieval with local context analysis. In: TOIS 2000, pp. 79–112 (2000)
22. van Rijsbergen, C.J.: Information Retireval. Butterworths, London (1979)