Web Object Retrieval*

Zaiqing Nie[1], Yunxiao Ma[2], Shuming Shi[1], Ji-Rong Wen[1], Wei-Ying Ma[1]

[1]Microsoft Research Asia, Beijing, China

[2]Peking University, Beijing, China

[1]{znie, shumings, jrwen, wyma}@microsoft.com, [2]mayx@infosec.pku.edu.cn

ABSTRACT

The primary function of current Web search engines is essentially relevance ranking at the document level. However, myriad structured information about real-world objects embedded in static Web pages and online Web databases. In this paper, we propose a paradigm shift to enable searching at the object level. In traditional information retrieval models, documents are taken as the retrieval units and the content of a document is considered reliable. However, this reliability assumption is no longer valid in the object retrieval context when multiple copies of information about the same object typically exist. These copies may be inconsistent because of diversity of Web site qualities and the limited performance of current information extraction techniques. In this paper, we propose several language models for Web object retrieval. We test these models on our academic search engine called Libra and compare their performances.

1. INTRODUCTION

The primary function of current Web search engines is essentially relevance ranking at the document level, a paradigm in information retrieval for more than 25 years. However, there are various kinds of objects embedded in static Web pages or Web databases. Typical objects are people, products, papers, organizations, etc. We can imagine that if these objects can be extracted and integrated from the Web, powerful object-level search engines can be built to meet users' information needs more precisely, especially for some specific domains. For example, by extracting a large set of product objects from Web data sources, when users search for a specific product, one can acquire a list of relevant product objects with clear information such as name, image, price, and features. Another example might be a search for research literature, where, with the concept of Web objects, the results could be a list of papers with explicit title, author, and conference proceedings. Such results are obviously more appealing than a list of URLS, which costs user's significant efforts to decipher for needed information. We believe object-level Web search is particularly necessary in building vertical web search engines such as product search (e.g. *Froogle*), people search, scientific Web search (e.g. *Google Scholar*, *CiteSeer*), job search, and so on. Such a perspective has led to significant research community interest, while related technologies such as data record extraction [6], attribute value extraction [15], and object identification on the Web [12] have been developed in recent years. These techniques have made it possible for us to extract and integrate all related Web information about the same object together as an information unit. We call these Web information units *Web objects*. Currently, little work has been done in retrieving and ranking relevant Web objects to answer user queries.

In this paper, we focus on exploring suitable models for retrieving Web objects. There are two direct categories of candidate models for object retrieval. The first is comprised of the traditional document retrieval models, in which all contents in an object are merged and treated as a text document. The other is made up of structured document retrieval models, where an object can be viewed as a structured document and objects attributed as different document representations, with relevance calculated by combining scores of different representations. We argue that simply applying both of these two categories of models on Web object retrieval does not achieve satisfactory ranking results. In traditional IR models, documents are taken as the retrieval units and the content of documents are considered reliable. However, the reliability assumption is no longer valid in the object retrieval context. There are several possible routes to introduce errors in object contents during the process of object extraction:

- **Source-level error:** Since the quality of Web sources can vary significantly, some information about an object in some sources may be simply wrong.
- **Record-level error:** Due to the huge number of Web sources, automatic approaches are commonly used to locate and extract the data records from Web pages or Web databases [6]. It is inevitable that the record extraction (i.e. detection) process will introduce additional errors. The extracted records may miss some key information or include some irrelevant information, or both.
- **Attribute-level error:** Even if the Web source is reliable and the object contents are correctly detected, the description of an object (i.e. object element labeling) may be still wrong because of incorrect attribute value extraction. For example, it is very common to label a product name by brand, or vice versa. In Citeseer, we usually find that author names are concatenated to paper titles, or some names are missing.

In this paper, we focus on this unreliability problem in Web object retrieval. Our basic ideas are based on two principles. First, as described above, errors can be introduced in both the record level and attribute level. Moreover, as errors will be propagated along the extraction process, the accuracy of attribute extraction is surely lower than that of record extraction. However, separating record contents into multiple attributes will bring more information than just treating all contents in a record as a unit. Therefore, it is desirable to combine both record-level representation and attribute-level representation. We hope, by combing representations of multiple levels, our method is insensitive to extraction accuracy.

* Microsoft Research Technical Report: MSR-TR-2006-70

Second, multiple copies of information about the same object usually exist. These copies may be inconsistent because of diverse Web site qualities and the limited performance of current information extraction techniques. If we simply combine the noisy and inaccurate object information extracted from different sources, we will not be able to achieve satisfactory ranking results. Therefore, we need to distinguish the quality of the records and attributes from different sources and trust data of high reliability more and data of low reliability less. We hope that even when data from some sites have low reliability, we can still get good retrieval performance if some copies of the objects have higher reliability. In other words, our method should also take advantage of multiple copies of one object to achieve stable performance despite varying qualities of the copies.

Based on the above arguments, our goal is to design retrieval models insensitive to data errors and that can achieve stable performance for data with varying extraction accuracies. Specifically, we propose several language models for Web object retrieval, namely a record-level representation model, an attribute-level representation model, and a model balancing record-level and attribute-level representations. We test these models on a paper search engine and compare their performance. We conclude that the best model is the one combining both record-level and attribute-level evidence and taking into account of the errors at different levels.

The rest of the paper is organized as follows. First, we define the Web object information retrieval problem. In Section 3, we introduce the models for Web object retrieval. In Section 4, we use a scientific Web search engine further motivate the Web object retrieval problem. After that, we report our experimental results in Section 5. Finally, we discuss related work in Section 6. Section 7 states our conclusion.

2. BACKGROUND AND PROBLEM DEFINITION

In this section, we first introduce the concept of Web objects and object extraction. We then define the Web object retrieval problem.

2.1 Web Objects and Object Extraction

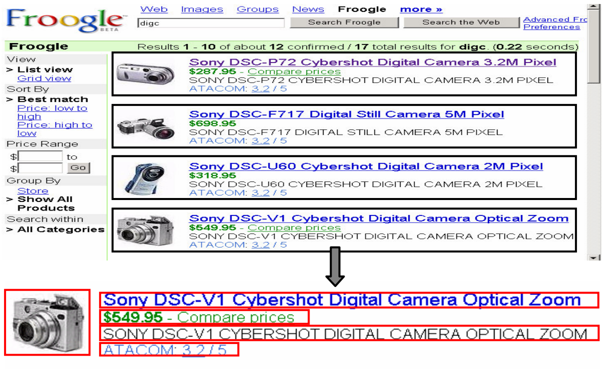
We define the concept of *Web Objects* as the principle data units about which Web information is to be collected, indexed, and ranked. Web objects are usually recognizable concepts, such as authors, papers, conferences, or journals that have relevance to the application domain. A Web object is generally represented by a set of attributes $A = \{a_1, a_2, ..., a_m\}$. The attribute set for a specific object type is predefined based on the information requirements in the domain.

If we start to think of a user's information need or a topic to search on the Web as a form of Web Object, the search engine will need to address at least the following technical issues in order to provide intelligent search results to the user:

- Object-level Information Extraction – A Web object is constructed by collecting related data records extracted from multiple Web sources. The sources for holding object information could be HTML pages, documents put on the Web (e.g. PDF, PS, Word, and other formats.), and deep contents hidden in Web databases. Figure 1 illustrates four data records embedded in a Web page and five attributes from a records. There is already extensive research to explore algorithms for extraction of objects from Web sources (more discussion about the diversity of sources is to come.)

- Object Identification and Integration – Each extracted instance of a Web object needs to be mapped to a real world object and stored into the Web data warehouse. To do so, we need techniques to integrate information about the same object and disambiguate different objects.

- Web object retrieval – After information extraction and integration, we should provide retrieval mechanism to satisfy users' information needs. Basically, the retrieval should be conducted at the object level, which means that the extracted objects should be indexed and ranked against user queries.



Sony DSC-P72 Cybershot Digital Camera 3.2M Pixel
$287.95 - Compare prices
SONY DSC-P72 CYBERSHOT DIGITAL CAMERA 3.2M PIXEL
ATACOM: 3.2 / 5

Sony DSC-F717 Digital Still Camera 5M Pixel
$698.95
SONY DSC-F717 DIGITAL STILL CAMERA 5M PIXEL
ATACOM: 3.2 / 5

Sony DSC-U60 Cybershot Digital Camera 2M Pixel
$318.95
SONY DSC-U60 CYBERSHOT DIGITAL CAMERA 2M PIXEL
ATACOM: 3.2 / 5

Sony DSC-V1 Cybershot Digital Camera Optical Zoom
$549.95 - Compare prices
SONY DSC-V1 CYBERSHOT DIGITAL CAMERA OPTICAL ZOOM
ATACOM: 3.2 / 5

Sony DSC-V1 Cybershot Digital Camera Optical Zoom
$549.95 - Compare prices
SONY DSC-V1 CYBERSHOT DIGITAL CAMERA OPTICAL ZOOM
ATACOM: 3.2 / 5

Figure 1. Four Data Records in a Web Page and Five Attributes from a Record

Figure 2 shows the compounds of a Web object and a flowchart to extract the object from Web sources. The key messages conveyed by the figure are:

1. The contents of a Web object are aggregated from multiple Web sources. These copies may be inconsistent because of the diverse Web site qualities and the limited performance of current information extraction techniques.

2. From each source, two steps are taken to extracted wanted information. First, record extraction [6] is applied to get data records relevant to the domain from the resource. Second, attribute extraction [15] is used to label different portions of each extracted record as different attributes. Both of the two steps are unlikely to be accurate. Record extraction can extract a totally wrong record, miss some parts of a record, or add irrelevant information to a record. Attribute extraction may wrongly label an attribute or not identify an attribute. But, in practice, the accuracy of every extraction algorithm on each Web source can be reasonably measured by using some test dataset. Therefore, we can assign the accuracy number to each extraction function in the figure and take it as a quality measurement of the data extracted. For record *k*, we use α_k to denote the accuracy of record detection, and γ_k to denote the accuracy of attribute extraction.

3. An object can be described at two different levels. The first one is the record-level representations, in which an object can be viewed as the collection of a set of extracted records and the attributes of each record are not further distinguished. The second on is the attribute-level representations, in which an object is made up of a

set of attributes and each attribute is a collection of attribute instances extracted from the records in multiple sources.

4. The importance of the j^{th} attribute, β_j , indicates the importance level of the attribute in calculating relevance probability. The problem of using differing weights for different attributes has been well studied in existing structured document retrieval work [8][7] and can be directly used in our Web object retrieval scenario.

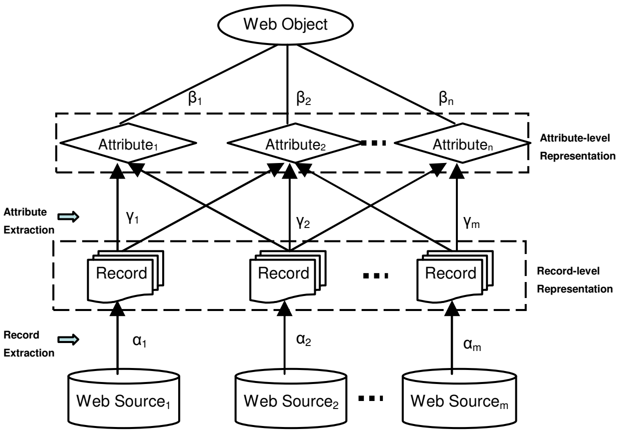


Figure 2. Web Object and Object Extraction

2.2 Web Object Retrieval

Our goal in this paper is to explore effective models to retrieval Web objects described above. The retrieval models should be insensitive to data errors and can achieve stable performance for data with varying extraction accuracies.

In document-level information retrieval, there is no concept of correctness. This is because there is no pre-defined semantic meaning of a document, and all the words and sentences in the document will define the meaning of the document. However the meaning of real world objects is pre-defined and the descriptions about the objects on the Web may be incorrect. Since the users usually want to see the correct information about the most relevant real-world objects first, it is critical to be able to use the accuracy of the extracted object descriptions in calculating the relevance probabilities of their corresponding real-world objects.

3. LANGUAGE MODELS FOR WEB OBJECT RETRIEVAL

In this section, we present a language model to estimate the relevance between an object and a query. We first provide background on language modeling for document retrieval. We then propose several language models for Web object retrieval.

3.1 Background on Language Modeling

Language models interpret the relevance between a document and a query as the probability of generating the query from the document's model. That is,

$$P(D \mid Q) \propto P(Q \mid D) \cdot P(D)$$

For a query Q, if independent among query terms are assumed, then it can be proved (by simple probability calculations) that,

$$P(Q \mid D) = \prod_{i=1}^{|Q|} P(w_i \mid D)$$

Where w_i is the i^{th} query term of Q, $|Q|$ is denoted as the length of Q, and $P(w_i|D)$ is the probability of generating term w_i from the language model of D.

Given word w and document D, by maximal likelihood estimation and Dirichlet smoothing, which is commonly used, the probability of generating term w by the language model of document D can be estimated as follows,

$$P(w \mid D) = \lambda \cdot \frac{tf(w,D)}{|D|} + (1-\lambda) \cdot \frac{tf(w,C)}{|C|} \qquad (3.1)$$

where $|D|$ is the length of document D, $tf(w,D)$ is the term frequency (i.e. number of terms) of term w in D, $|C|$ is the number of terms in the whole collection, and $tf(w,C)$ is the term frequency of term w in the whole collection C. λ can be treated as a parameter with its value in [0, 1]. It is common to let λ rely on document length $|D|$, as follows,

$$\lambda = \frac{|D|}{|D| + \mu}$$

where μ is a parameter and it is common to set it according to the average document length in the collection.

3.2 Web Object Retrieval

In the following subsections, we present language models for Web object retrieval.

3.2.1 Record-level Representation Model

One simple way of scoring a Web object against a query is to consider each record as the minimum retrieval unit. In this way, all the information within a record is considered as a bag of words without further differentiating the attribute values of the object, and we only need to know the accuracy of record extraction. The advantage of this model is that no attribute value extraction is needed, so we can avoid amplifying the attribute extraction error for some irregular records whose information can not be accurately extracted.

If we consider all the information about an object as a big document consisting of K records, we can have a language model for each record and combine them, as [7] have been done. One approach to combining the language models for all the records of object o is as follows,

$$p(w \mid o) = \sum_{k=1}^{K} \left(\alpha_k P(w \mid R_k) \right)$$

where $P(w|R_k)$ is the probability of generating w by the record R_k, and α_k is the accuracy of record extraction.

$P(w|R_k)$ can be computed by treat each record R_k as a document. Therefore, by using Formula 3.1, we have,

$$P(w \mid R_k) = \lambda \frac{tf(w,R_k)}{|R_k|} + (1-\lambda) \frac{tf(w,C)}{|C|}$$

Where C is the collection of all the records.

In this model, we only need to know the record extraction accuracy which can be easily obtained through empirical evaluation. Note that the parameters α_k are normalized accuracy numbers and $\sum_k \alpha_k = 1$.

The intuition behind this model is that we consider all the fields within a record equally important and give more weight to the correctly detected records.

3.2.2 Attribute-level Representation Model

For the object records with good extraction patterns, we do hope to use the structural information of the object to estimate relevance. It has been shown that if we can correctly segment a document into multiple weighted fields (i.e. attributes), we can achieve more desirable precision [8][7]. In order to consider the weight difference of difference fields and avoid amplifying the attribute extraction error too much, we need to consider attribute extraction accuracy.

We consider all the information about an object as a big document consisting of *K* records and each record has *M* fields (i.e. attributes), and use the formula below to estimate the probability of generating term w by the language model of object O ,

$$P(w \mid O) = \sum_{k=1}^{K}\left(\alpha_k \gamma_k \sum_{j=1}^{M} \beta_j P(w \mid O_{jk})\right)$$

Where $\alpha_k\gamma_k$ together can be considered as the normalized accuracy of both record detection and attribute extraction of record k , and $\sum_k \alpha_k \gamma_k = 1$. β_j is the importance of the j^{th} field, and $\sum_j \beta_j = 1$. Here $P(w|O_{jk})$ is the probability of generating *w* by the j^{th} field of record *k*. $P(w|O_{jk})$ can be computed by treating each O_{jk} as a document (Formula 3.1 is used again here),

$$P(w \mid O_{jk}) = \lambda \cdot \frac{tf(w, O_{jk})}{\mid O_{jk} \mid} + (1-\lambda) \cdot \frac{tf(w, C_j)}{\mid C_j \mid}$$

Where C_j is the collection of all the j^{th} fields of all the objects in the object warehouse.

The intuition behind this formula is that we give different weights to individual fields and give more weight to the correctly detected and extracted records.

3.2.3 Model Balancing Record-level and Attribute-level Representations

As we discussed earlier, the record-level representation method has the advantage of handling records with irregular patterns at the expenses of ignoring the structure information, while attribute-level representation model can take the advantage of structure information at the risk of amplifying extraction error.

We argue that the best way of scoring Web objects is to use the accuracy of extracted object information as the parameter to find the balance between structured and unstructured ways of scoring the objects. We use the formula below to estimate the probability of generating term w by the language model of object O ,

$$P(w \mid O) = \sum_{k=1}^{K}\left(\alpha_k \sum_{j=1}^{M}\left(\gamma_k \beta_j + (1-\gamma_k)\frac{1}{M}\right) P(w \mid O_{jk})\right)$$

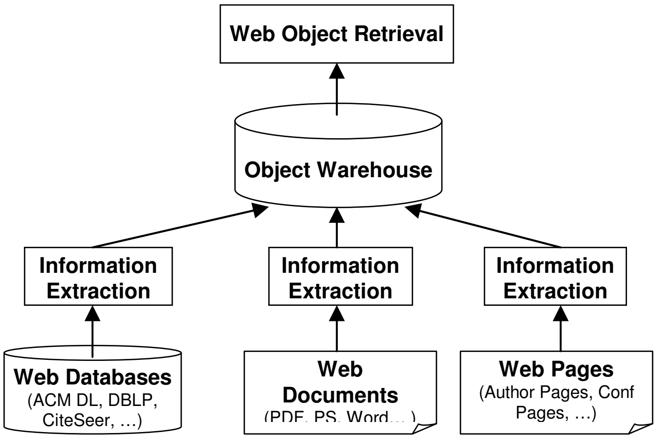


Figure 3. Libra Architecture

The basic intuition behind this formula is that we give different weights to individual fields for correctly extracted records and give the same weight to all the fields for the incorrectly extracted records.

4. A Case Study

Below we will use *Libra* [16][17], an academic search engine we have built to motivate the Web object retrieval problem.

As shown in Figure 3, we extract information from different Web databases and pages to build structured databases of Web objects including researchers, scientific papers, conferences, and journals. The objects can be retrieved and ranked according to their relevance to the query. The relevance is calculated based on all the collected information about this object, which is stored with respect to each individual attribute. For example, research paper information is stored with respect to the following attributes: title, author, year, conference and abstract. In this way, we can also handle structured queries and give different weights to different attributes when calculating relevance scores. Compared with *Google Scholar* and *CiteSeer*, both of which solely search paper information at the document level, this new engine can retrieve and rank other types of Web objects. This includes authors, conferences and journals with respect to a query. This greatly benefits junior researchers and students in locating important scientists, conferences, and journals in their research fields.

We focus on exploring suitable models for retrieving Web objects. We argue that simply applying tradition document-level IR models on Web object retrieval will not be able to achieve satisfactory ranking results. In traditional IR models, document is taken as the retrieval unit and the content of a document is reliable. However the reliability assumption is no longer valid in the object retrieval context. Multiple copies of information about the same object usually exist, and such copies may be inconsistent because of diverse Web site qualities and the limited performance of current information extraction techniques. If we simply combine the noisy and inaccurate attribute information extracted from different sources, we may not be able to achieve satisfactory ranking results.

5. EVALUATION

The goal of the evaluation is to show that the best way of scoring Web objects is to combine representations of multiple levels, when the object information is collected from multiple inconsistent data sources. Although there're some developed test collections in IR fields, such as TREC, INEX etc, there is little work on retrieving information from multiple inconsistent sources, and we can not find any publicly available collections (datasets) for evaluation. For this reason, we evaluate the work in the context of *Libra,* the real paper search engine we developed.

5.1 Datasets

In the experiments, we use four structured sources: SCI, DBLP, ACM Digital Library, and CiteSeer. In addition to the above four structured sources, we also integrate information extracted from an unstructured source including all the PDF files crawled from the Web. Totally we have information about 1.4 million computer science papers integrated from the five data sources.

For the unstructured source, we use a PDF to HTML converter to automatically convert the PDF files into HTML files, and then extract the information such like paper title, author, abstract, and references for each paper, by a paper extractor we developed. During the system development process, we developed three versions of paper extractors with varying accuracy levels named PEv1, PEv2 and PEv3 for short. We empirically evaluated the extraction accuracy for these extractors, and the PEv3 achieved the best score, the PEv2 was less acceptable while the lowest was PEv1.

5.2 Query Set

We selected some queries from the log of *Libra* according to the following criteria:

- The frequent query has high priority to be selected.
- All the queries about author name, conference/journal name, or year will be removed. Because our model only returns the document contains all the query terms, and it's very likely that the retrieved document is relevant to the query if only such kind of query term existed in it. Then no significant differences could be observed between models.
- Specific queries with few answers will be removed.

At last, we select 79 queries as the query set.

5.3 Retrieval Models

We implemented two other simple retrieval models in addition to the three models we introduced in Section 3, and observed their precisions in our experiments.

- Bag of Words (**BW**): We treat all term occurrences in a record equally and there is no difference between records either. This is actually the traditional document retrieval model that considers all the information about the same object as a bag of words. Indeed, this is a special case for the record-level representation model that each the record is assigned the equal α_k .
- Record-level Representation model (**RR**): This model is the described in Section 3.2.1. Comparing to the BW model, this model takes the accuracy of record detection into account.
- Multiple Weighted Fields (**MWF**): This method assigns a weight to each attribute (β_j) and amends the $P(w|O_{jk})$ by multiplying the weight of the corresponding attribute. However, it does not consider the extraction error. We use the same α_k and γ_k for all records in the attribute-level representation model for this model.

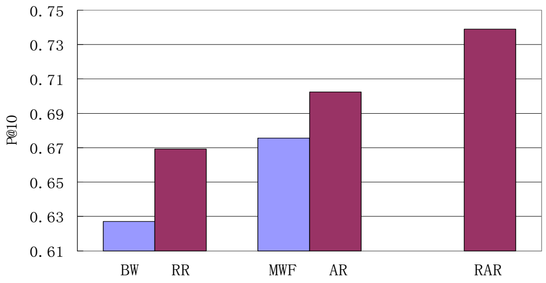


Figure 6. Precision at 10

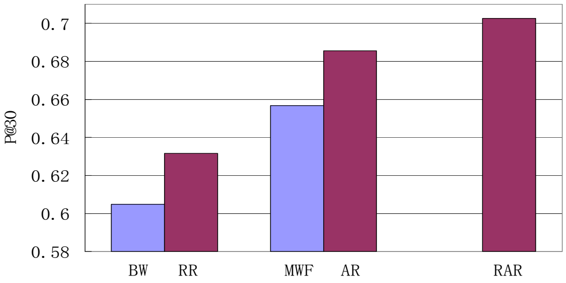


Figure 7. Precision at 30

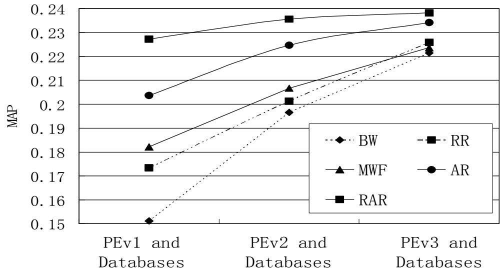


Figure 8. Average Precision (MAP) with Different Quality Data Sources

- Attribute-level Representation model (**AR**): This model is the described in Section 3.2.2.
- Balancing Record-level and Attribute-level Representations (**RAR**): This model is described in Section 3.2.3.

5.4 Parameter Setting

Compared to the traditional unstructured document retrieval, in our model we set a weight of each attribute (β_j). The weights of the attributes are tuned manually by considering the importance of attributes. The extraction accuracy (α_k and γ_k) of each data source is set by sampling some data from each data source.

5.5 Experimental Results

For each query, we collected the top 30 results from each algorithm and labeled the relevance level of each paper. In order to ensure a fair labeling process, all the top papers from all the models were shuffled before they were sent to the labeler. Because our queries belonged to several domains, like database, web search, and security, we asked persons who have different backgrounds to handle the queries they were familiar with. We observed the precision at 10, precision at 30 and average precision (MAP) of all five models. The result clearly shows that the RAR model balancing record-level and attribute-level representations is consistently better than other models.

In Figure 6 we show the precision at rank=10 of the results returned by the five retrieval models, and we show the precision at rank=30 of the results returned by the five retrieval models in Figure 7. As we can see, the models that considered accuracy levels of the extractors have better precision, and the RAR model is significantly better than the other models. This is especially true if we want to reduce the error for the top ranked results (for example, at rank=10).

We believe that even though several low quality data sources were used, we can achieve good retrieval results by combining all evidence from all data sources. To verify this, each time we used one of our developed extractors (PEv1, PEv2, and PEv3), and the four web databases (ACM, Citeseer, DBLP, SCI) to complete our experiments, the quality of PEv1, PEv2 and PEv3 become better and better. The MAP results for the five models are shown in Figure 8. Because the results of P@10 and P@30 are similar to the MAP, we omitted them. The result clearly illustrates that the RAR model is almost insensitive to noise from low quality data sources if we use the evidence from other data sources, and our RAR model is rather robust. In addition, models that consider extraction accuracy levels are consistently better than comparative models. Finally, the gap between models that consider extraction accuracy and models that not consider extraction accuracy will increase when noise increases.

6. RELATED WORK

In recent years, researchers began to segment web pages into blocks [2][6] to promote retrieval precision in web search. In block retrieval works, researchers primarily care about the way of segmenting web pages, and usually use the highest relevance score of a block as the score of whole page. There are also many studies on structured document retrieval [10][5] and utilizing multiple fields of web pages for web page retrieval [7][9]. These methods linearly combine the relevance score of each field to solve the problem of scoring structured documents with multiple weighted fields. In [8], the authors show that the type of score linear combination methods is not as effective as the linear combination of term frequencies. In our work, we follow this way of handling the multiple attributes problems.

XML retrieval has attracted great interest recently. Many works have been done to solve the wide variety of length among XML elements [13], and to deal with the overlap problem[14]. Besides these issues, people have also developed test collections like INEX. But because there's no extraction process, all the retrieval units of a document are from the same XML, they do not handling data inconsistency issues.

We noticed that a need exists for document-level Web page retrieval to handle the anchor text field of a page, which is extracted from multiple Web pages [3]. Researchers in this area often treat all of the anchor texts as a bag of words for retrieval. There is little work which considers the quality of extracted anchor text. Moreover, since anchor text is a single field independently extracted from multiple Web pages, there is no need for structured retrieval.

The work on distributed information retrieval [4][11] is related to our work in the sense that it combines information from multiple sources to answer user queries. However, other researchers focus on selecting the most relevant search engines for queries and rank query results instead of integrating object information.

The final rank of an object is determined by both its popularity and its relevance to the query. [1][16] is focused on the former and the goal of this paper is to calculate the latter.

7. CONCLUSION

There is lots of structured information about real-world objects embedded in static Web pages or online Web databases. Our work focuses on object level retrieval, which is a completely new perspective, and differs significantly from the existing structured document retrieval and passage/block retrieval work. We propose several language models for Web object retrieval, test these models on our *Libra* academic search engine and compare their performances. We conclude that the RAR model is the superior by taking into account the extraction errors at varying levels.

8. REFERENCES

[1] A. Balmin, V. Hristidis and Y. Papakonstantinou. Authority-Based Keyword Queries in Databases using ObjectRank**.** VLDB, 2004.

[2] D. Cai, S. P. Yu, J-R Wen and W-Y Ma. Block-based Web Search. SIGIR, 2004.

[3] R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin and David P. Williamson. Searching the Workplace Web. WWW, 2003.

[4] L. Gravano and H. Garcia-Molina. Generalizing gloss to vector-space databases and broker hierarchies. VLDB, 1995.

[5] M. Lalmas. Dempster-Shafer's Theory of Evidence Applied to Structured Documents: Modeling Uncertainty. SIGIR, 1997.

[6] B. Liu, R. Grossman, and Y. H. Zhai. Mining Data Records in Web Pages. SIGKDD, 2003.

[7] P. Ogilvie and J. Callan. Combining document representations for known item search. SIGIR, 2003.

[8] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. CIKM, 2004.

[9] T. Westerveld, W. Kraaij and D. Hiemstra. Retrieving Web Pages using Content, Links, URLs and Anchors. TREC2001, 2001.

[10] R. Wilkinson. Effective Retrieval of Structured Documents. SIGIR, 1994.

[11] J. Xu, and J. Callan. Effective retrieval with distributed collections. SIGIR, 1998.

[12] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. SIGKDD, 2002.

[13] J. Kamps, M. de Rijke and B. Sigurbjörnsson. Length normalization in XML retrieval. SIGIR, 2004.

[14] C. L.A. Clarke. Controlling Overlap in Content-Oriented XML Retrieval. SIGIR, 2005

[15] J. Zhu, Z. Nie, J-R. Wen, B. Zhang, W-Y. Ma. 2D Conditional Random Fields for Web Information Extraction. ICML, 2005.

[16] Z. Nie, Y. Zhang, J-R. Wen, W-Y. Ma. Object-Level Ranking: Bringing Order to Web Objects. WWW, 2005.

[17] Libra. An Object-level Academic Search Engine. http://libra.directtaps.net