Web Object Retrieval

Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, Wei-Ying Ma
Microsoft Research Asia, Beijing, China
{znie, yunxiaom, shumings, jrwen, wyma}@microsoft.com

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 is embedded in static Web pages and online Web databases. Document-level information retrieval can unfortunately lead to highly inaccurate relevance ranking in answering object-oriented queries. 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 However, this reliability assumption is no longer valid in the object retrieval context when multiple copies of information about the same object typically exist.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. If we simply combine the noisy and inaccurate attribute information extracted from different sources, we may not be able to achieve satisfactory retrieval performance. In this paper, we propose several language models for Web object retrieval, namely an unstructured object retrieval model, a structured object retrieval model, and a hybrid model with both structured and unstructured retrieval features. We test these models on a paper search engine and compare their performances. We conclude that the hybrid model is the superior by taking into account the extraction errors at varying levels.

Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval – Retrieval Models

General Terms

Algorithms, Experimentation

Keywords

Web Objects, Information Retrieval, Language Model, Information Extraction

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 [1]. 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 [26]. For example, in our *Windows Live Product Search* project (http://products.live.com), we automatically extract a large set of product objects from Web data sources [38], 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. We have been developing another object-level vertical search system call *Libra Academic Search* (http://libra.msra.cn) to help researchers and students locate information for scientific papers, authors, conferences, and journals. With the concept of Web objects, the search results of *Libra* 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, people search, scientific Web search, job search, community search, and so on. Such a perspective has led to significant research community interest, while related technologies such as data record extraction [21][32][22], attribute value extraction[37], and object identification on the Web [31] 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 the object attributes 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 [22]. 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.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

- **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 also usually find that author names are concatenated to paper titles, or some author names are missing.

Although [38] proposed a model which combined the record and attribute extraction processes, it may also bring both record and attribute level error which are similar to other technique. 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. 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 an unstructured object retrieval model, a structured object retrieval model, and a hybrid model with both structured and unstructured retrieval features. We test these models on a paper search engine and compare their performance. We conclude that the best model is the one combining both object-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 need for object-level Web search and its advantages and challenges over existing search engines. After that, we report our experimental results in Section 5. Finally, we discuss related work in Section 6. Section 7 states our conclusions.

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 requirements in the domain.

If we start to think of a user 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 six data records embedded in a Web page and six 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.

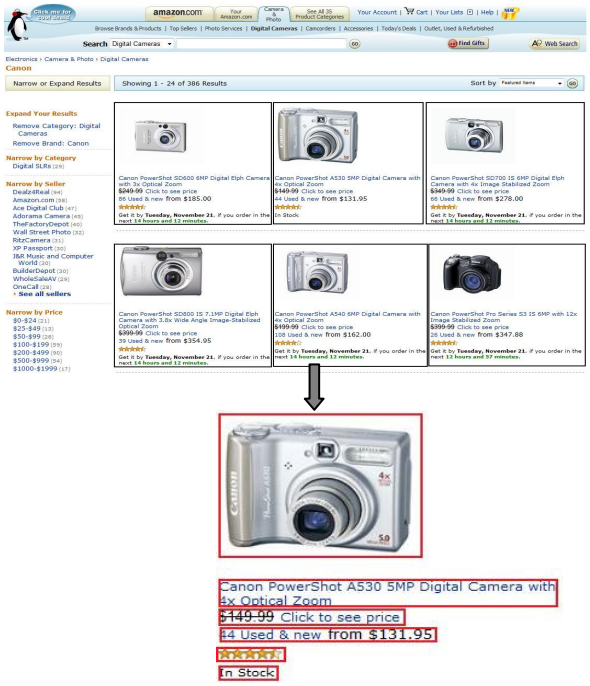

Figure 1. Six Data Records in a Web Page and Six Attributes from a Record

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 extract the wanted information. First, record extraction [21][32][22] is applied to get data records relevant to the domain from the resource. Second, attribute extraction [37] 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. We use α_k to denote the accuracy of record detection, and γ_k to denote the accuracy of attribute extraction of record k.
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 [30][28] and can be directly used in our Web object retrieval scenario.

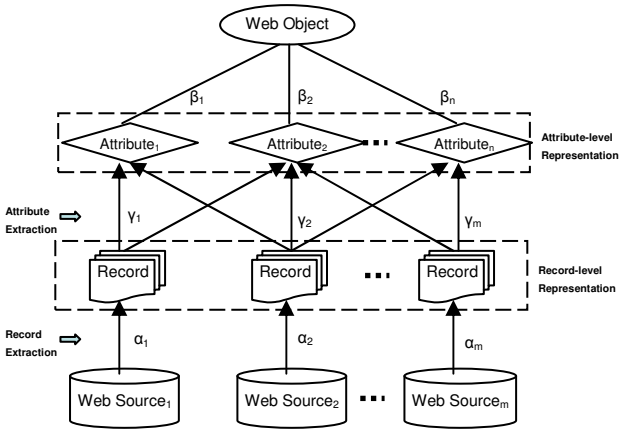

Figure 2. Web Object and Object Extraction

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 accuracy.

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, namely an unstructured object retrieval model, a structured object retrieval model, and a hybrid model with both structured and unstructured retrieval features.

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 independence 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, maximum likelihood estimation (MLE) is commonly used to estimate probability $P(w|D)$. Smoothing, which adjusts term probabilities to overcome data sparseness, is critical to the performance of language models. Among various smoothing methods, the Dirichlet prior smoothing is frequently discussed. By maximum likelihood estimation and Dirichlet smoothing, the probability of generating term w by the language model of document D can be estimated as follows,

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

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. In the above formula, 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 cannot be accurately extracted. This model can also be called unstructured object retrieval model since it treats each record as an unstructured document.

Now we present a language model for record-level Web object retrieval. 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 [28] have been done. One approach to combining the language models for all the records of object o is as follows,

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

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,

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

Where C is the collection of all the records, and λ is set according to Dirichlet prior smoothing.

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 [30][28]. In order to consider the weight difference of different fields and avoid amplifying the attribute extraction error too much, we need to consider attribute extraction accuracy. This model can also be called structured object retrieval model since it treats each record as a structured document.

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 we use the formula below to estimate the probability of generating term w by the language model of object o,

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

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,

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

Where C_j is the collection of all the j^{th} fields of all the objects in the object warehouse, and λ is set according to Dirichlet prior smoothing.

The intuition behind this formula is that we give different weight 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 unstructured object retrieval method has the advantage of handling records with irregular patterns at the expenses of ignoring the structure information, while attribute-level retrieval method 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,

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

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 (http://libra.msra.cn), a working scientific Web search engine we have built to motivate the need for object-level Web search and its advantages and challenges over existing search engines.

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, abstract, and full text. 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.

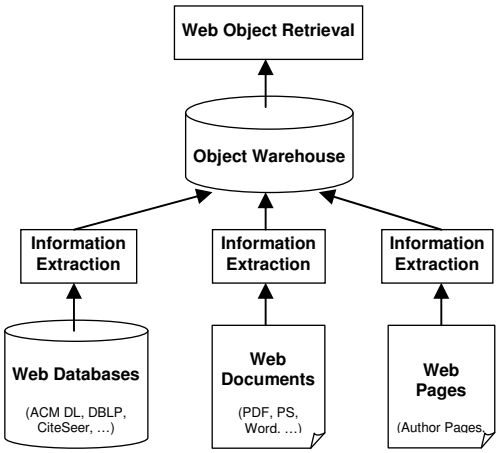

Figure 3. An Object-level Search Engine for Scientific Web

We focus on exploring suitable models for retrieving Web objects. We argue that simply applying traditional 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. For example, in Table 1 we show the title and author information about a paper from *DBLP* and *CiteSeer*. As can be seen, the information from *DBLP* is almost correct because it is manually input. However, the information from *CiteSeer* is noisy and inaccurate because it is automatically extracted.

Table 1. Inconsistency Example

Source	Title	Authors
Ground Truth	Towards Higher Disk Head Utilization: Extracting Free Bandwidth From Busy Disk Drives	Christopher R. Lumb, Jiri Schindler, Gregory R. Ganger, David Nagle, Erik Riedel
CiteSeer	Towards Higher Disk Head Utilization:	Extracting Free Bandwidth From Busy Disk Drives Christopher R. Lumb, Jiri...
DBLP	Towards Higher Disk Head Utilization: Extracting "Free" Bandwidth from Busy Disk Drives	Christopher R. Lumb, Jiri Schindler, Gregory R. Ganger, David Nagle, Erik Riedel

5. EVALUATION

The goal of the evaluation is to show that the best way of scoring Web objects is balancing structured and unstructured retrieval method, 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 cannot find any publicly available collections (datasets) for evaluation. For this reason, we evaluate the work in the context of Libra.

5.1 Datasets

Libra contains 1.4 million computer science papers extracted from Web databases, pages and file. Libra integrates papers information by their titles, authors and years from Web databases such as DBLP, ACM Digital Library, CiteSeer and SCI.

In addition to papers from Web databases, Libra also crawls papers which are in PDF format from the Web. After the files are crawled, we use some program to convert the PDF files into HTML files, and then extract the following attribute information: paper title, author, abstract, and references, by an extractor we developed. During the system development process, we developed three versions of the extractor, which we named PEv1, PEv2 and PEv3 for short. Since the main purpose of our experiments is to study the effectiveness of our model in handling varying extraction errors, we choose these three extractors from three different stages of our development process with varying accuracy levels for both record detection and attribute extraction. 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 (see Table 3 for their extraction accuracy numbers).

Although we save many attributes of the paper object, only title, author and abstract are used in our experiments. Because most data sources provide such info and they're the key elements to determine the relevance of a paper given a query.

To measure the inconsistent problem, we use the vector space model (VSM) [1] to calculate the distances between data sources. Since DBLP does not provide abstracts of papers, we compute the title distances between all the data sources as well as the full document (title, author and abstract) distances between sources exclude DBLP. Each document is represented by a VSM model and compared with its corresponding document in other data source. Then we calculate the average distances between data sources. The pairwise results could be seen in Table 2, from which, it's clear to see that the inconsistent problem is rather serious even for short text like title.

5.2 Query Set

We select some queries from one year 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 are 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.
- The queries that are too specific are removed. For example, a query like 'The PageRank Citation Ranking: Bringing

Order to the Web' which is a title of paper will get only two documents that contains all the query terms.

- The selected queries are examined by the researchers in our organization to make sure that they have unambiguous meaning.

At last, we get 79 queries as the query set belonging to several domains, like database, Web search and security etc.

Table 2. Pairwise VSM Distances between Data Sources
(S1=Source1, S2=Source2, D1=Distance Score of Titles, D2=Distance Score of Full Documents)

S1	S2	D1	D2		S1	S2	D1	D2
Citeseer	ACM	0.77	0.70		DBLP	SCI	0.94	-
Citeseer	DBLP	0.83	-		DBLP	PEv1	0.68	-
Citeseer	SCI	0.81	0.72		DBLP	PEv2	0.72	-
Citeseer	PEv1	0.73	0.63		DBLP	PEv3	0.78	-
Citeseer	PEv2	0.76	0.61		SCI	PEv1	0.70	0.62
Citeseer	PEv3	0.74	0.68		SCI	PEv2	0.71	0.65
ACM	DBLP	0.92	-		SCI	PEv3	0.76	0.68
ACM	SCI	0.88	0.85		PEv1	PEv2	0.83	0.76
ACM	PEv1	0.65	0.56		PEv1	PEv3	0.71	0.70
ACM	PEv2	0.67	0.60		PEv2	PEv3	0.76	0.73
ACM	PEv3	0.74	0.66					

5.3 Retrieval Models

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

- Bag of Words (**BW**): In this model, 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.
- Unstructured Object Retrieval (**UOR**): This is the record-level representation model 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 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.
- Structured Object Retrieval (**SOR**): This model is the attribute-level representation model described in Section 3.2.2.
- Balancing Structured and Unstructured Retrieval (**BSUR**): 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. To determine the extraction accuracy (α_k and γ_k), we sampled some data for each data source, then compute the accuracy for both record and attribute extraction results. Table 3 shows the results.

Table 3. Extraction Accuracy Parameters

	Citeseer	ACM	DBLP	SCI	PEv1	PEv2	PEv3
α_k	0.80	0.92	0.96	0.94	0.68	0.69	0.76
γ_k	0.74	0.95	0.97	0.91	0.63	0.73	0.78

Although ACM, DBLP and SCI are built manually and got high extraction accuracy, we can't totally depend on them to ensure data coverage. For example, the ACM only provides about 300,000 papers and many important articles are not covered. In addition, to keep the up to date data, the search engine has to crawl PDFs from the Web and extract info in them. Therefore, we have to utilize information from every source. Because each source provides only a subset of the papers in *Libra*, no single data source can dominate the results.

5.5 Experimental Results

For each query, we try the five models over all the infomation from 7 data sources (DBLP, ACM Digital Library, CiteSeer, SCI, PEv1, PEv2 and PEv3). Then the top 30 results of every query are collected from each algorithm and labeled with relevance judgments. In order to ensure a fair labeling process, all the top papers from all the models are merged before they were sent to the labeler. In this way the labeler could not know the ranked position and the connection between the models and the ranking results. We ask labelers with different background to handle the queries they are familiar with. We observe the precision at 10, precision at 30, average precision (MAP) and the precision-recall curve to measure the performance of all five models. The result clearly shows that the Balancing Structured and Unstructured Retrieval (BSUR) model is consistently better than other models.

In Figure 4 we show the precision at rank=10 of the results returned by the five retrieval models, in Figure 5 we show the precision at rank=30 of the results returned by the five retrieval models, and Figure 6 is the average precision (MAP) for all the five models. The Precision-Recall curve is also plotted in Figure 7. As we can see, the models that considered accuracy levels of the extractors have better precision, and the BSUR model is much 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).

In addition to the performance test, statistical tests are also used to determine the significance of differences [16][36]. We did the paired t-test analysis on F1 score. After grouped models with insignificant performance, the p-value shows that BSUR is significantly better than {UOR, MWF, SOR} which are significantly better than BW.

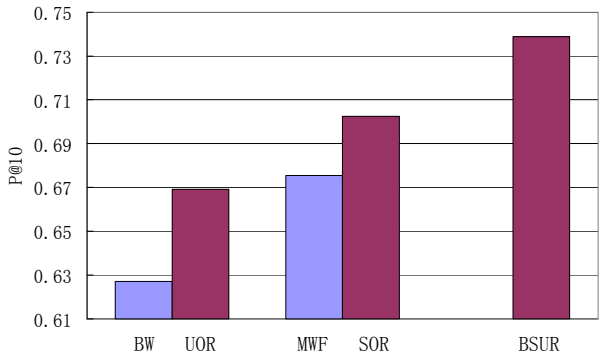

Figure 4. Precision at 10

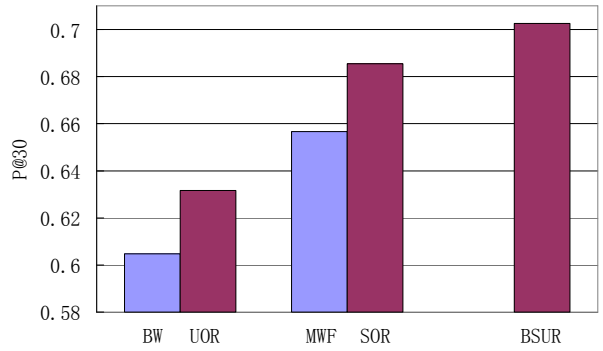

Figure 5. Precision at 30

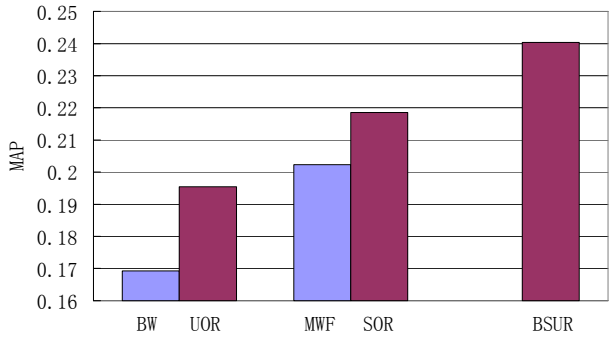

Figure 6. Average Precision (MAP)

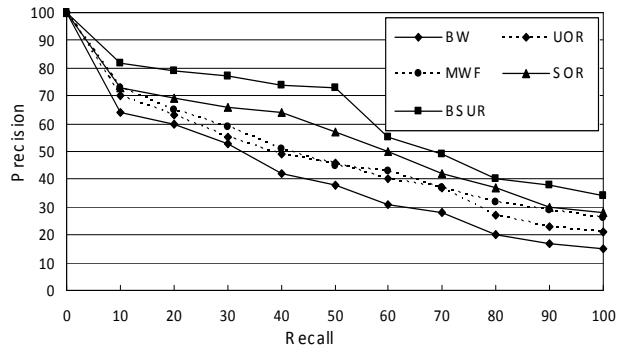

Figure 7. Precision at 11 Standard Recall Levels

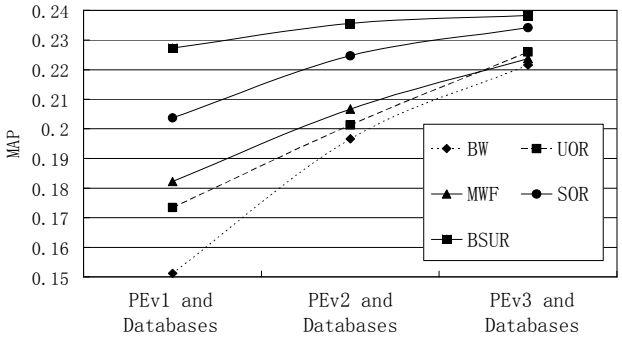

Figure 8. Average Precision (MAP) with Different Quality Data Sources

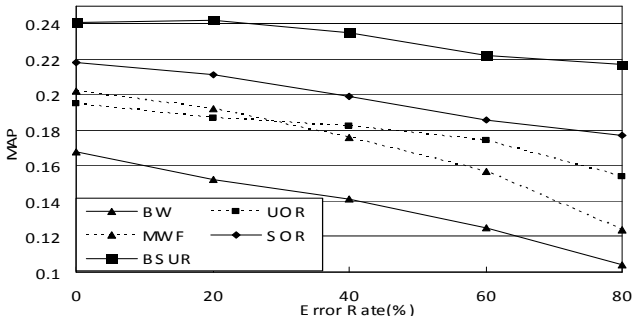

Figure 9. Average Precision (MAP) with Different Error Rate

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 use 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 results, we omitted them. The result clearly illustrates that the BSUR model is almost insensitive to noise from low quality data sources if we use the evidence from other data sources, and our BSUR 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.

To better control the error rate of data, we also manually add noise into the dataset. Both of record and attribute level errors of a record are brought in by adding irrelevant words, discarding some words or exchanging words between attributes according to some desired error rate. In this experiment, we introduce noise into ACM and SCI dataset, because they provide full documents data with best quality**.** The accuracy of these sources are set based on the error rate. Figure 9 shows the MAP results of all the models with different error rates. Because there is much more noise, the improvement and robustness of the model considering data qualities are much more significant.

6. RELATED WORK

Two types of ranking are considered in Web search engines, static rank and dynamic rank. For object level vertical search, the static rank has been studied in [1]. In this paper, we focus on the calculation of dynamic object rank.

There has been much work on passage retrieval [4][18] in the traditional document retrieval area. In recent years, researchers began to segment Web pages into blocks [22][2][3] to promote retrieval precision in Web search. In passage retrieval or block retrieval works, researchers primarily care about the way of segmenting documents or Web pages, and usually use the highest relevance score of a passage or block as the score of whole document or page. There are also many studies on structured document retrieval [34][19] and utilizing multiple fields of Web pages for Web page retrieval [28][33][9][6]. These methods linearly combine the relevance score of each field to solve the problem of scoring structured documents with multiple weighted fields. In [30], the authors show that the type of linear score 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 problem.

XML retrieval has attracted great interest in recent years because of document presentation in XML form providing opportunities to utilize the structure of documents. Many works have been done to handle query language [13] for XML, solve the wide variety of length among XML elements [17], and to deal with the overlap problem – one tag may be contained by another tag [7]. 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 handle data inconsistency issues.

However, our work focuses on object level retrieval, which is much closer to users' requirements and considers the quality of each data source and the accuracy of the extracted object information during retrieval. This is a completely new perspective, and differs significantly from the structured document retrieval and passage/block retrieval work we discussed above.

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 [8][10]. 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 unstructured retrieval. Because ignoring the structure information will not help improving the quality of the anchor text, there is no need for balancing structured and unstructured retrieval models.

The work on distributed information retrieval [5][14][23][35] 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.

Information quality is one of the most important aspects of Web information integration, and it is closely related to our work since we need to know the quality of the data sources. Many interesting techniques have been studied on estimating the quality of the Web sources and databases [25][24].

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, namely an unstructured object retrieval model, a structured object retrieval model, and a hybrid model with both structured and unstructured retrieval features. We test these models on *Libra Academic Search* and compare their performances. We conclude that the hybrid model is the superior by taking into account the extraction errors at varying levels.

8. REFERENCES

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley Publishers, 1999.

[2] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. Block-Level Link Analysis. In Proceedings of SIGIR, 2004.

[3] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. Block-based Web Search. In Proceedings of SIGIR, 2004.

[4] J. P. Callan. Passage-Level Evidence in Document Retrieval. In Proceedings of SIGIR, 1994.

[5] J.P. Callan. Distributed information retrieval. In Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, edited by W. Bruce Croft. Kluwer Academic Publisher, pp. 127-150, 2000.

[6] Abdur Chowdhury, Mohammed Aljlayl, Eric Jensen, Steve Beitzel, David Grossman and Ophir Frieder. Linear Combinations Based on Document Structure and Varied Stemming for Arabic Retrieval. In The Eleventh Text REtrieval Conference (TREC 2002), 2003.

[7] Charles L.A. Clarke. Controlling Overlap in Content-Oriented XML Retrieval. In Proceedings of the SIGIR, 2005.

[8] Nick Craswell, David Hawking and Stephen Roberson. Effective Site Finding using Link Anchor Information. In Proceedings of SIGIR, 2001.

[9] Nick Craswell, David Hawking and Trystan Upstill. TREC12 Web and Interactive Tracks at CSIRO. In The Twelfth Text Retrieval Conference(TREC 2003), 2004.

[10] Ronald Fagin, Ravi Kumar, Kevin S. McCurley, Jasmine Novak, D. Sivakumar, John A. Tomlin and David P. Williamson. Searching the Workplace Web. In Proceedings of the Twelfth International World Wide Web Conference, 2003.

[11] Hui Fang, Tao Tao and ChengXiang Zhai. A Formal Study of Information Retrieval Heuristics. In Proceedings of SIGIR, 2004.

[12] Norbert Fuhr. Probabilistic Models in Information Retrieval. The computer Journal, Vol.35, No.3, pp. 243-255.

[13] Norbert Fuhr and Kai Großjohann. XIRQL: A Query Language for Information Retrieval in XML documents. In Proceedings of the SIGIR, 2001.

[14] L. Gravano and H. Garcia-Molina. Generalizing gloss to vector-space databases and broker hierarchies. In Proceeding of the International Conference on Very Large Data Bases (VLDB), 1995.

[15] Jiawei Han and Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmman Publishers, 2000.

[16] David Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. In Proceedings of the ACM SIGIR, 1993.

[17] Jaap Kamps, Maarten de Rijke and Börkur Sigurbjörnsson. Length normalization in XML retrieval. In Proceedings of the SIGIR, 2004.

[18] M. Kaszkiel and J. Zobel. Passage Retrieval Revisited. In Proceedings of SIGIR, 1997.

[19] Mounia Lalmas. Dempster-Shafer's Theory of Evidence Applied to Structured Documents: Modeling Uncertainty. In Proceedings of SIGIR, 1997.

[20] Mounia Lalmas, Uniform representation of content and structure for structured document retrieval. Technical Report, Queen Mary and Westfield College, University of London, 2000.

[21] K. Lerman, L. Getoor, S. Minton, and C. A. Knoblock. Using the structure of Web sites for automatic segmentation of tables. In ACM SIGMOD Conference (SIGMOD), 2004.

[22] Bing Liu, Robert Grossman, and Yanhong Zhai. Mining Data Records in Web Pages. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2003.

[23] M. Meng, K. Liu, C. Yu, W. Wu, and N. Rishe. Estimating the usefulness of search engines. In ICDE Conference, 1999.

[24] Amihai Motro and Igor Rakov. Estimating the quality of databases. In Proceedings of the 3rd International Conference on Flexible Query Answering (FQAS), Roskilde, Denmark, May 1998. Springer Verlag.

[25] Felix Naumann and Rolker Claudia. Assessment Methods for Information Quality Criteria. In Proceedings of the International Conference on Information Quality (IQ), Cambridge, MA, 2000.

[26] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen and Wei-Ying Ma. Object-Level Ranking: Bringing Order to Web Objects. In Proceedings of the 14th international World Wide Web Conference (WWW), 2005.

[27] Zaiqing Nie, Ji-Rong Wen and Wei-Ying Ma. Object-Level Vertical Search. To appear by the Third Biennial Conference on Innovative Data Systems Research (CIDR), 2007.

[28] Paul Ogilvie and Jamie Callan. Combining Document Representations for known item search. In Proceedings of SIGIR, 2003.

[29] S. E. Robertson, S. Walker, S. Jones and M. M. Hancock-Beaulieu. Okapi at TREC-3. In The Third Text REtrieval Conference (TREC 3), 1994.

[30] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 Extension to Multiple Weighted Fields. ACM CIKM, 2004.

[31] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In Knowledge Discovery and Data Mining (KDD), 2002.

[32] J. Wang and F. H. Lochovsky. Data extraction and label assignment for Web databases. In World Wide Web conference (WWW), 2003.

[33] Thijs Westerveld, Wessel Kraaij and Djoerd Hiemstra. Retrieving Web Pages using Content, Links, URLs and Anchors. In The Tenth Text REtrieval Conference (TREC2001), 2001.

[34] Ross Wilkinson. Effective Retrieval of Structured Documents. In Proceedings of SIGIR, 1994.

[35] J. Xu, and J. Callan. Effective retrieval with distributed collections. In Proceedings of SIGIR, 1998.

[36] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In Proceedings of the ACM SIGIR, 1999.

[37] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, Wei-Ying Ma. 2D Conditional Random Fields for Web Information Extraction. In Proceedings of the 22nd International Conference on Machine Learning (ICML), 2005.

[38] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, Wei-Ying Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2006.