Object-level Web Information Retrieval*

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

[1]Microsoft Research Asia, Beijing, China

[2]Peking University, Beijing, China

[1]{t-znie, 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, there is lots of structured information about real-world objects embedded in static Web pages and online Web databases. Document-level information retrieval will unfortunately lead to highly inaccurate relevance ranking in answering object-oriented queries. In this paper, we consider a new paradigm shift to enable searching at the object level. In traditional information retrieval 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 However the reliability assumption is no longer valid in the object retrieval context where usually exist multiple copies of information about the same object. These copies may be inconsistent because of the 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 will not be able to achieve satisfactory retrieval performance. In this paper, we introduce a probabilistic model to handle the inconsistency problem using the source quality information, and our empirical evaluation shows that our object-level model is significantly better than the existing document-level models.

Keywords

Web Objects, Information Retrieval, Probabilistic Model, Information Extraction

1. INTRODUCTION

Today search engines have become one of the most critical applications on the Web, driving many important online businesses that connect users to information. As the Web continues to grow its size with a variety of new data and penetrate into every aspect of our life, the need for developing a more intelligent search engine is increasing.

The primary function of current Web search engines is essentially relevance ranking at the document level, an old paradigm in information retrieval for more than 25 years [1]. We believe in 2-5 years this type of general Web search technologies will become commodity. In this paper, we are considering a new paradigm shift to enable searching at the object level. For example, when a user is looking for information about a researcher, the user is not interested in just retrieving a set of papers or documents in which this researcher's name appears, but more interested in finding insight and knowledge about what works he has done in different period of time, what important contributions he has brought to the research community, how influential he is, and his social network, etc. Such kind of intelligence is not possible to obtain through current web search engines.

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 – Information (e.g. important attributes) about a Web object is usually distributed in many Web sources and within small segments of Web pages. To extract the information, we need techniques to detect the Web page segments (i.e., object blocks and elements that will be discussed in details later) and then label each element in the object block accordingly.
- Object Identification and Integration – Each extracted instance of 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.
- Object-level Web information 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 to user queries.

We believe object-level Web search is particularly necessary in building vertical web search engines such as product search (e.g. *Froogle* [31]), people search, scientific Web search (e.g. *Google Scholar* [32], *CiteSeer* [29]), job search, community search, and so on.

Below we will use *OSSE (real name omitted for double-blind reviewing purposes),* a 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.

Motivating Example: As shown in Figure 1, we extract and integrate 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 attribute information about this object on the Web. For example, paper information is stored with respect to the following attributes: title,

* Technical Report. Microsoft Research. **MSR-TR-2005-11**

* This work is done when the author is visiting Microsoft Research Asia.

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 the relevance score. Compared with *Google Scholar* and *CiteSeer* which solely search paper information at the document level, this new engine can retrieve and rank other types of Web objects such as 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 field.

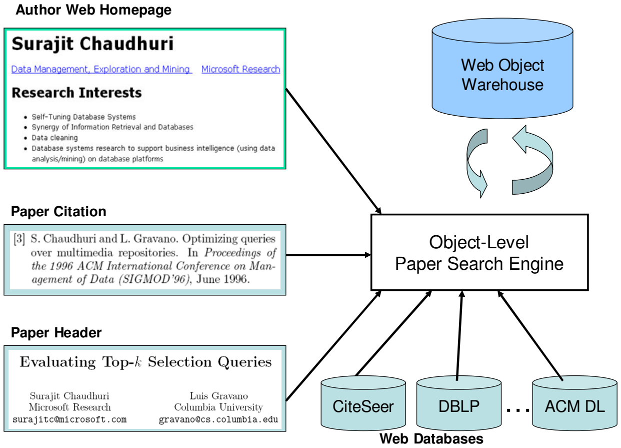

Author Web Homepage
Surajit Chaudhuri
Data Management, Exploration and Mining
Microsoft Research
Research Interests
Self-Tuning Database Systems
Synergy of Information Retrieval and Databases
Data cleaning
Database systems research to support business intelligence (using data analysis/mining) on database platforms
Paper Citation
[3] S. Chaudhuri and L. Gravano. Optimizing queries over multimedia repositories. In Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD'96), June 1996.
Paper Header
Evaluating Top-k Selection Queries
Surajit Chaudhuri
Microsoft Research
surajitc@microsoft.com
Luis Gravano
Columbia University
gravano@cs.columbia.edu
Web Object Warehouse
Object-Level Paper Search Engine
CiteSeer
DBLP
...
ACM DL
Web Databases

Figure 1. An Object-level Search Engine for the Scientific Web.

In this paper, 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. There usually exist multiple copies of information about the same object. These copies may be inconsistent because of the 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 will 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* [30] 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. The unreliability of objects is caused by the following reasons:

- **Unreliable data sources:** As we show in the above example, the quality of the Web sources can vary significantly, and some information about an object may be simply wrong.
- **Incorrect object detection:** Since we need to locate and extract the block of a Web page containing information about an object [18], it is inevitable that the object extraction process will introduce additional errors.
- **Incorrect attribute value extraction:** Even if the Web source is reliable and the object block is correctly detected, the description of an object (i.e. object element labeling) may be still wrong because of incorrect attribute value extraction.

In this paper, we focus on this unreliability problem in the object-level Web information retrieval. We use the quality score for each Web source as the confidence of the extracted object information. Specifically we introduce an object description generation model to explain how to estimate the correct description of an object extracted from multiple inconsistent Web sources.

We also propose several Web object retrieval models: unstructured object retrieval model, structured object retrieval model, and a hybrid model with both structured and unstructured retrieval features. We conclude that the best way of scoring Web objects is to use confidence score of the extracted information as the parameter to find the balance between the structured and unstructured way of scoring the objects. This is because the unstructured object retrieval method has the advantage of handling blocks with irregular patterns at the expenses of ignoring the structure information, while structured retrieval method can take the advantage of structure information at the risk of amplifying the extraction error.

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

The rest of the paper is organized as follows. Next we define the Web object information retrieval problem. In Section 3, we introduce the models for Web object retrieval. After that we report our experimental results in Section 4. Finally we discuss the related work in Section 5 and conclude the paper in Section 6.

2. PROBLEM DEFINITION

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

2.1 Web Objects & Attributes

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 which have relevance to the application domain. Different types of objects are used to represent the information for different concepts. We assume the same type of objects follows a common relational schema: $R(a_1, a_2, ..., a_m)$.

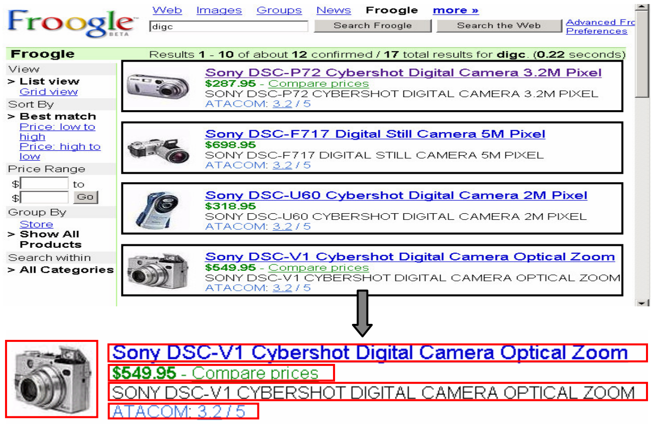
Attributes $A = \{a_1, a_2, ..., a_m\}$ are properties which describe the objects. There are three types of object attributes:

- Key attributes: properties which can uniquely identify an object,
- Important attributes: distinctive properties other than the key attributes,
- Others attribute: all the other properties

The designer of the system needs to determine the types of objects which are relevant to the application, and the key and important attributes of these objects.

2.2 Object Blocks & Elements

The information about an object on a Web page is usually grouped together as a block, since Web page creators are always trying to display semantically related information together. Using explicit or implicit visual separators such as lines, blank area, image, font, and color, we can first locate these object blocks based on existing Web page segmentation technologies like [2]. Figure 2 shows that four object blocks are located in a Web page generated by Froogle.



Results **1** - **10** of about **12** confirmed / **17** total results for **digc**. (**0.22** seconds)

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

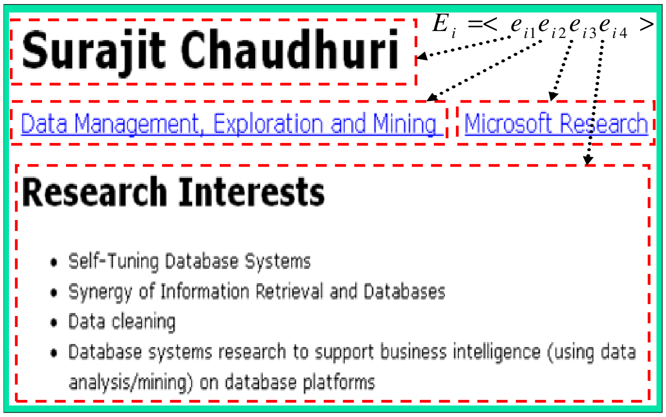
Sony DSC-F717 Digital Still Camera 5M Pixel
$698.95
SONY DSC-F717 DIGITAL STILL CAMERA 5M PIXEL
ATACOM: 3.2 / 5

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 2. Four Object Blocks in a Web Page



Surajit Chaudhuri

Data Management, Exploration and Mining | Microsoft Research

Research Interests

- Self-Tuning Database Systems
- Synergy of Information Retrieval and Databases
- Data cleaning
- Database systems research to support business intelligence (using data analysis/mining) on database platforms

Figure 3. An example object block and its elements from a computer scientist homepage. Four object elements are located.

With the help of data record mining [18] and classification [14] techniques, we can then automatically determine whether the object blocks are relevant to the application. However automated object block detection and classification is beyond the scope of this paper, where we assume that the relevant blocks are given.

Given an object block found on a Web page, it is straightforward to further segment it to atomic extraction entities using the visual information and delimiter, such as font, position, color, appearance pattern, and punctuation. Figure 3 shows an example object block with four atomic extraction entities, which are called *object elements*. Each element only belongs to an attribute of the object, and an attribute can contain several elements.

There have been many existing technologies such as [35, 36] have been proposed to further extract the attribute values based on template discovery in HTML codes. Moreover, object identification techniques [37] are then used to integrate all the labeled attribute values from various Web sources about the same object into a single information unit.

2.3 Web Object Retrieval

Figure 4 shows that a Web object with multiple attributes and each attribute may contain information from multiple object blocks. The importance of the i^{th} attribute, imp_i, indicates the importance level of the attribute in calculating the relevance probability. The problem of using difference weights for different attributes has been well studied in existing structured document retrieval work [24] and can be directly used in our Web object retrieval scenario. The confidence level, $conf_i$, of the object block i is dependent on the quality of its information source and the accuracy of the wrapper which is used to detect the object blocks and extract the attribute values. As we discussed in the previous section, the existence of inconsistent attribute values is the major difference between object-level information retrieval and traditional document-level retrieval.

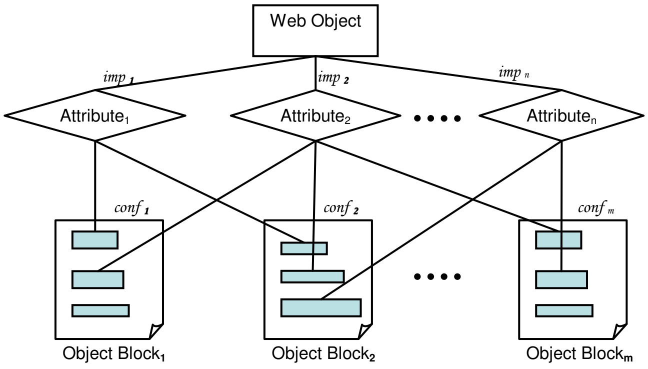


Figure 4. Web Object and Object Blocks

Given a Web object O and query Q, we aim to calculate the probability $P(R \mid O)$, i.e. the probability that the object O is relevant to Q given that it has the correct description about the real world object.

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 confidence of the extracted object descriptions in calculating the relevance probabilities of their corresponding real-world objects.

This paper studies the problem of handling the inconsistency issues in calculating the object relevance probability. Specifically, we introduce a novel object description generation model to estimate the frequencies of all terms in the correct description of a Web object, and use the confidence of the extracted object information from a block to decide the weight of structure information of the block to be used in calculating the term frequency.

3. MODELS FOR WEB OBJECT RETRIEVAL

In this section, we propose several models to estimate the frequency of a term appearing in the correct description of a Web object. We first introduce an object description generation model to explain how to estimate the frequency of the term in a Web object extracted from multiple inconsistent and unstructured blocks. We then propose several models to convert structured blocks with multiple weighted fields into unstructured blocks which are the input of the object description generator.

3.1 Object Description Generator

We now introduce a model which automatically generates the Web object description from a set of object blocks extracted from the Web (see Figure 5). We use the description generation model to explain how to estimate the term frequency of a Web object extracted from multiple inconsistent object blocks.

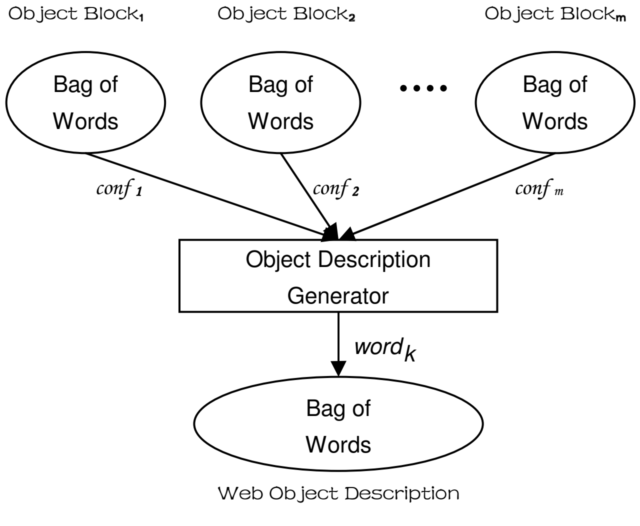


Figure 5. Object Description Generator

Let's use BW_i to denote the set of distinctive words in the i^{th} object block. For each $word_k \in \bigcup_i BW_i$, the description generator will look for the word from the bags of all the blocks. The generator will decide whether to put the word into the bag of the Web object by calculating the probability that it is truly a word of the object. Then the generator will remove one occurrence of the word from the bags containing the word, and check whether these bags have more occurrences of the word, until no more occurrences are found. We calculate the frequency of the word in the following way:

$$tf_k = \sum_{t=1}^{\max\{tf_{k1}, tf_{k2}, \ldots, ft_{km}\}} \left[P\left(\bigcup_{\{j | tf_{kj} - t >= 0\}} E_{kjt} \right) \right]$$

Where E_{kjt} denotes the event that the t^{th} occurrence of the $word_k$ from the bag of the j^{th} block is truly an occurrence of the word in the bag of the Web object. We can use the confidence, $conf_j$, of the object block to approximate the probability, $P(E_{kjt})$, of E_{kjt}. As we discussed before, the confidence level of an object block is source and wrapper dependent, and it's a prior knowledge which can be estimated empirically or predicted mathematically.

In the following subsections, we introduce several models to weight the frequencies of the terms in each structured object block before we apply the generation model to calculate the term frequency of each word in the object.

3.2 Unstructured Object Retrieval

One simple way of scoring a Web object against a query is to consider each object block as the minimum retrieval unit, and use our object description generation model to estimate the term frequency of each query term. In this way, all the information within an object block is considered as a bag of words without further differentiating the attribute values of the object, and we only need to know the confidence level for each object block. The advantage of this method is that no attribute value extraction is needed, so we can avoid amplifying the extraction error for some irregular blocks whose information can not be accurately extracted. We use the object description generation model to get an approximate of the term frequency of each word by combining the corresponding term frequencies in each block using its confidence level.

3.3 Structured Object Retrieval

For the object blocks with good extraction patterns, we do hope to use the structural information of the object to estimate its relevance. It has been shown that if we can correctly segment a document into multiple weighted fields, we can get much better precision [24].

In [24], the authors show the effectiveness of converting a structured document with multiple weighted fields into an unstructured document, by simply combining the term frequencies of the different fields by forming a linear combination weighted by their weights. Similarly we can convert a structured object block into an unstructured block, and then apply the block-level object retrieval method described in the above subsection to get an approximate of term frequency in the object.

Let's use tf_{kmj} to denote the term frequency of $word_k$ in $attribute_j$ of $block_m$, w_j to denote the weight of $attribute_j$, and then we can calculate tf_{km} in the following way:

$$tf_{km} = \sum_j w_j tf_{kmj}$$

3.4 Balancing Structured and Unstructured Retrieval

As we discussed earlier, the block-level unstructured object retrieval method has the advantage of handling blocks 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 the extraction error.

We argue that the best way of scoring Web objects is to use the confidence of the extracted object information as the parameter to find the balance between the structured and unstructured way of scoring the objects.

Let's use tf_{kmj} to denote the term frequency of $word_k$ in $attribute_j$ of $block_m$, w_j to denote the weight of $attribute_j$, $conf_m$ to denote the confidence of the information extraction from $block_m$, tf'_{km} to denote the term frequency of $word_k$ in $block_m$ obtained by considering the object block as an unstructured block, and then we can calculate tf_{km} in the following way:

$$tf_{km} = (1 - conf_m)tf'_{km} + conf_m \sum_j w_j tf_{kmj}$$

4. EVALUATION

The goal of the evaluation is to show that the best way of scoring Web objects is to the balancing structured and unstructured retrieval method, when the object information is collected from multiple inconsistent data sources. Since there is little work on retrieving information from multiple inconsistent sources, we can not find any publicly available collections (datasets) for evaluation. For this reason, we evaluate the work in the context of *OSSE,* the paper search engine we developed.

4.1 Datasets

We select 25 frequent queries from the query log of *OSSE,* which contains 1 million computer science papers extracted from Web databases and pages, as the test query set. *OSSE* integrates paper information from three Web databases: DBLP, ACM Digital Library [28], and CiteSeer. In order to evaluate the retrieval effectiveness of the method on object collected from multiple sources, we purposely select only the objects whose information is extracted from all the three sources for evaluation. The paper titles are used as the key to uniquely identify by a paper object. In addition to the *OSSE* sources, we also extract paper information from two other online Web databases CSB [34] and IEEE Xplore [33] by querying them using the 25 test queries:

> *data mining*, *web search*, *information retrieval*, *data*, *web classification*, *data integration*, *machine learning*, *association rules*, *image retrieval*, *design pattern*, *database*, *data cluster*, *mobile web search*, *video retrieval*, *volume rendering*, *minimum cut*, *block web*, *web mining*, *Hierarchical clustering*, *web security*, *web data extraction* computer vision, Application level multicast, personalized Web search, information retrieval language model

All together we collected 19866 objects, among them there are 15629 objects that have more than three sources.

4.2 Noisy Data

The quality of the extracted object information is determined by the quality of the corresponding information sources, object block detection accuracy, and attribute value extraction accuracy. Since the quality of the sources is fixed. We can only change the block detection and attribute extraction accuracy to add noisy data. The object block detection error will cause some attributes containing information from other blocks, and the attribute extraction error will cause some attributes containing information which belongs to other attributes in the same object block. For example, as explained in table 1, the author information from *Citeseer* contains some words from the title. In order to simulate the two types of errors we mentioned, we add noisy data to the *CSB* and *IEEE* datasets in the following way:

- During the crawling process, our wrapper will extract inaccurate abstract of a paper in certain probability. The inaccurate data may be a subset of the true abstract, or a superset containing information from other fields. We set the error rate of *CSB* to 40% and *IEEE* to 30%.
- After the crawling process, we randomly choose the block detection error rate from 10% to 80% for the papers from *CSB* and exchange their abstract to introduce error across blocks.

4.3 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 the term occurrences in an object block equally and there is no difference between blocks either. This is actually the traditional document retrieval model which considers all the information about the same object as a bag of words.
- Unstructured Object Retrieval (**UOR**): This is the model described in section 3.2. Comparing to the BW model, this model takes the confidence level of each block into account.
- Multiple Weighted Fields (**MWF**): This method assigns a weight to each attribute and amend the frequency of a term occurrence by multiplying the weight of the corresponding attribute of the occurrence.
- Structured Object Retrieval (**SOR**): This model is described in section 3.3.
- Balancing Structured and Unstructured Retrieval (**BSUR**): This model is described in section 3.4, and we consider it as best model.

We use a simplified BM25 [24] as the ranking function for the backend retrieval system. It is of the form

$$\sum_{T\in Q}\frac{(k_1+1)d_j}{k_1((1-b)+b\frac{dl}{avdl})+d_j}\log\frac{N-df_j+0.5}{df_j+0.5}$$

Where df_j is the document frequency of term j, dl is the document length, $avdl$ is the average document length across the collection, and k_1 and b are free parameters. We set $k1 = 2.0$ and $b = 0.25$ for all the retrieval models. For each model, d_j is the amended term frequency and dl is the sum of frequency of every term. Because we treat all the words of an object as a document, the document frequency of term j is identical in every model.

4.4 Parameter Setting

Comparing to the traditional unstructured document retrieval, we have two additional parameters to be set: the weight of each attribute and the confidence of each source. The weights of the attributes are tuned manually by considering the importance of attributes, while the confidence of data source is predefined. The wrapper will be changed to add more noisy data as the confidence level of the source decreases. To be fair, we fix the values of parameters for all the models.

4.5 Experimental Results

For each query, we collect the top 30 results of each algorithm and label the relevance of each paper. In order to ensure a fair labeling process, all the top papers from all the models are mixed together before they are sent to the person doing the labeling. In this way the person can't know the specific ranking position and the connection between the models and the ranking results. We observe the precision at 10 and precision at 30 of all the 5 models. The result clearly shows that the BSUR (Balancing Structured and Unstructured Retrieval) model is significantly and 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 in Figure 7 we show the precision at rank=30 of the results returned by the five retrieval models, and the noisy data is added according to the block detection error rate of CSB as 20%, and attribute extraction error rate of CSB as 40% and IEEE Xplore as 30%. As we can see the all the models considering the confidence level of the extracted object information have better precision, this is especially true if we observe more ranking results (for example at rank=30), and BSUR model is significantly better than other models, this is especially true if we want to reduce the error for the top ranked results (for example at rank=10).

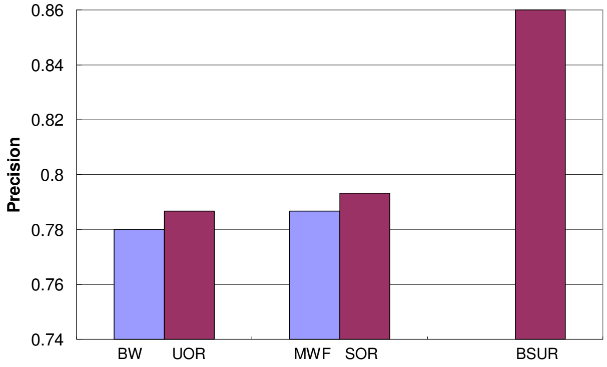


Figure 6. Precision at 10

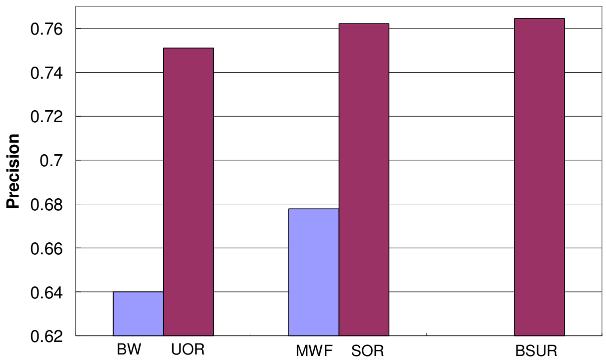


Figure 7. Precision at 30

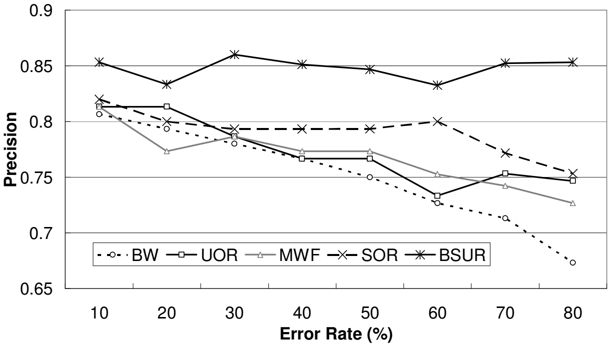


Figure 8. Precision at 10 with Different Error Rate

In Figure 8 we show the precision variations at rank=10 for all models as we change the block detection error rate of CSB data source from 10% to 80%. The result clearly illustrates that 1). The BSUR (Balancing Structured and Unstructured Retrieval) model is almost insensitive to the noise from a low quality data source when there are many high quality data sources; 2) the models which consider the confidence level of the extracted information are consistently better than its comparative models; 3) the gap between models considering confidence and models not considering confidence will increase when the noise increases; 4) when the noise is not very strong, the MWF (multiple weighted fields) model is better than the UOR (unstructured object retrieval) model which considers confidence but treating all the words in a block equally. This is because the MWF model assigns larger weight to the title field which equals to duplicate the words in the title field, and we did not purposely introduce error into the title field, so the heavier weight in the correct extracted title information can neutralize the negative effect caused by noise in the abstract field.

5. RELATED WORK

There has been much work on passage retrieval [4, 15] in traditional document retrieval area. In recent years, researchers began to segment web pages into blocks [18, 2, 3] to promote the retrieval precision in web search. In the passage retrieval or block retrieval works, researchers primarily care the way of segmenting documents or web pages, and usually simply use the highest relevance score of a passage or block as the score of whole document or page. There are also lots of works on structured document retrieval [26, 16] and utilizing multiple fields of web pages for web page retrieval [22, 25, 8, 6]. These methods linearly combine the relevance score of each field to solve the problem of scoring structured documents with multiple weighted fields. In [24], 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 problem.

However, our work focuses on object level retrieval which is much closer to users' requirements, and considers the quality of each data source 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 document-level Web page retrieval also need to handle the anchor text field of a page which is extracted from multiple Web pages [7, 10]. The 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 the extracted anchor text. Moreover, since the anchor text is a single field extracted from multiple Web pages independently, there is no need for unstructured retrieval. Because ignoring the structure information will not help improving the quality of the anchor text. So there is no need for balancing structured and unstructured retrieval model.

The works on distributed information retrieval [5, 13, 19, 27] are related to our work in the sense of combining information from multiple sources to answer user queries. However they focus on selecting the most relevant search engines for the query and ranking their query results instead of integrating the 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 confidence level of the extracted object information. Many interesting techniques have been studied on estimating the quality of the Web sources and databases [21, 20]. We can leverage these techniques in compute the confidence of the extracted object information.

6. CONCLUSION AND FUTURE WORK

There is lots of structured information about real-world objects embedded in static Web pages or online Web databases. Extracting, integrating and retrieving the information about the same Web object will enable us to build more powerful and intelligent search engines. This paper studies the problem of Web object retrieval. In particular, we introduce a Web object generation model to estimate the correct description of a Web object, and propose a novel retrieval model to score the Web object with respect to a user query. The model takes into account the confidence of the extracted object information and finds the balance point between structured and unstructured retrieval in calculating the relevance score of a Web object. The experimental results show that our Web object retrieval is significantly and consistently better than traditional retrieval models.

As we pointed out, the unreliability of objects is caused by unreliable data sources, incorrect object region detection and incorrect attribute value extraction. In this paper, we mainly focus on dealing with unreliable data sources and incorrect attribute value extraction problems. In the next step, we will include the factor of incorrect object region detection in the framework and make the model more resistant to the variety of data quality.

7. 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. *ACM SIGIR Conference (SIGIR),* 2004.

[3] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. Block-based Web Search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[4] J. P. Callan. Passage-Level Evidence in Document Retrieval, In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Informa-tion Retrieval*, 1994.

[5] J. 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] Nick Craswell, David Hawking and Stephen Roberson. Effective Site Finding using Link Anchor Information. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

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

[9] David Cramel, Yoelle S. Maarek, Matan Mandelbrod, Yosi Mass, and Aya Soffer. Searching Xml Documents via Xml Fragments. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.

[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 the 27th Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval, 2004.

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

[13] 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.

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

[15] M. Kaszkiel and J. Zobel. Passage Retrieval Revisited. In *Proceedings of the twentieth Annual International ACM SIGIR Conference on Research and Development in Infor-mation Retrieval*, 1997.

[16] Mounia Lalmas. Dempster-Shafer's Theory of Evidence Applied to Structured Documents: Modeling Uncertainty. In *Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1997.

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

[18] 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.

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

[20] 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.

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

[22] Paul Ogilvie and Jamie Callan. Combining document repre-sentations for known item search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.

[23] 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.

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

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

[26] Ross Wilkinson. Effective Retrieval of Structured Documents. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[27] J. Xu, and J. Callan. Effective retrieval with distributed collections. In *Proceeding of the ACM SIGIR Conference* (SIGIR), 1998.

[28] ACM Digital Library. *Http://portal.acm.org/dl.cfm*

[29] CiteSeer**:** Scientific Literature Digital Library. *Http://citeseer.ist.psu.edu*.

[30] DBLP computer science bibliography. *Http://www.informatik.uni-trier.de/~ley/db*.

[31] Froogle. *Http://froogle.google.com*.

[32] Google Scholar. *Http:// scholar.google.com.*

[33] IEEE Xplore. *Http://ieeexplore.ieee.org.*

[34] The Collection of Computer Science Bibliographies. Http://liinwww.ira.uka.de/bibliography.

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

[36] 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.

[37] 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.