A System for Indexing Tables, Algorithms and Figures

Pradeep Teregowda
Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
pbt105@cse.psu.edu

Madian Khabsa
Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
mkhabsa@psu.edu

C. Lee Giles
Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802
giles@ist.psu.edu

ABSTRACT

Indexing objects such as documents, figures, tables and algorithms in a single system presents challenges in schema mapping, detecting overlapping objects in documents, presenting results from such an system to users. We propose a federated approach to indexing and retrieving such objects in academic papers.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; H.3.1 [**Content Analysis and Indexing**]: Indexing methods

Keywords

Indexing, Embedded Objects, Information Retrieval Systems

1. INTRODUCTION

Objects embedded in academic documents such as figures, tables, equations and algorithms provide valuable metadata in the form of results, data, pseudocode and relationships. Extraction and indexing of such objects has been studied in the context of [4] and algorithms [1]. In these cases, the indexing and ranking are specific to particular objects and features specific to the objects. CiteSeer^x [6] [1] provides an example, allowing search for such objects using multiple independent indices and interfaces.

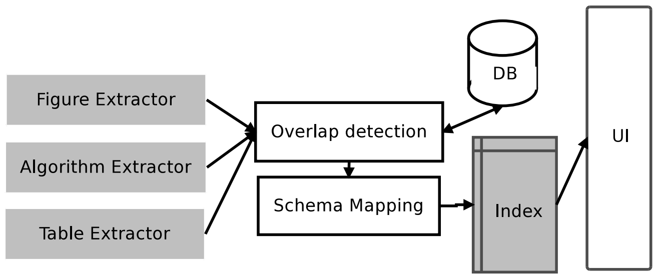
Such an approach is highly inefficient to the user, and information relevant to a particular topic could be distributed across several indices. Many approaches to solving this exist and have been studied as problems of diversity in retrieving documents [5] and subtopic retrieval [7]. We propose a system built by indexing diverse objects obtained from academic documents into a single index which can be queried by the user. The interactive query interface enables users to drill down into particular object types.

2. PROPOSED SYSTEM

The architecture of the proposed system is show in Figure 1. It includes extractors, overlap detection coupled with an index and an user interface.

[1]`http://citeseerx.ist.psu.edu`

Figure 1: System Architecture



We utilized previously existing extractors for tables [4] and algorithms [1]. For figure mention extraction we built a extractor using PDFBox with a rule based system (derived from a CART model). We link objects to documents utilizing a checksum or a processing id. The metadata available from these extractors are shown in table 1. We choose document metadata (title, citations, id), object type and the most common elements (in bold in table 1) for indexing. In case of missing fields, equivalent fields were identified for the object type. Objects extracted from a document might be labeled as algorithms and as figures or tables. We built a detection system for detecting such overlaps with noisy metadata using near duplicate detection methods. It uses a similarity measure for object content of two objects, selected by blocking using document id and object id.

Empirically, utilizing a threshold of 0.66 for similarity with the Jaccard similarity measure for captions produced good results.

3. INDEXING AND RANKING

In comparison to building individual index and customized ranking for each object, building a single index is more complex. One of the prominent challenges, is to infer the object type from an user query to effectively satisfy user needs. In this context we identify the relevant object type for queries by utilizing existing query logs and keywords relevant to particular object types and finally the object content itself [3, 2].

We make use of Apache Solr [2] an open source search server, for indexing objects. Using citation based ranking methods is the probably the first choice such a method would not be able to rank all objects due to limitations in the ci-

[2]`http://lucene.apache.org/solr/`

Field	Tables	Figures	Algorithms
Page number	×	×	×
Caption	×	×	×
Contents	×		
Footnote	×		
Reference text	×	×	×

Table 1: Object Schema

tation graph coverage. For example only 32% of the documents in the collection are cited in CiteSeer^x.

Instead we modify user queries by adding weights to queries based on identified keywords. Keywords for boosting for indexing tables were were obtained after cleaning the existing table index in CiteSeer^x. In the case of algorithms, we made use of keywords extracted from a list of algorithms from Wikipedia and text books on data structures and algorithms.

For cases where a keyword occurs in both lists of previous query terms from logs and keywords, we weight both object types equally.

4. RETRIEVAL

A query submitted to the user interface is processed by the application to identify keywords, the query is then weighed and submitted to the index. Results returned are displayed in the web interface. The interface allows users to view results from each object type by selecting the object type tab. Each search result includes the caption, reference text and document title. The particular document page on which the embedded object is placed can be viewed by clicking on the search result. The query interface allows the user to

Figure 2: User Search Interface

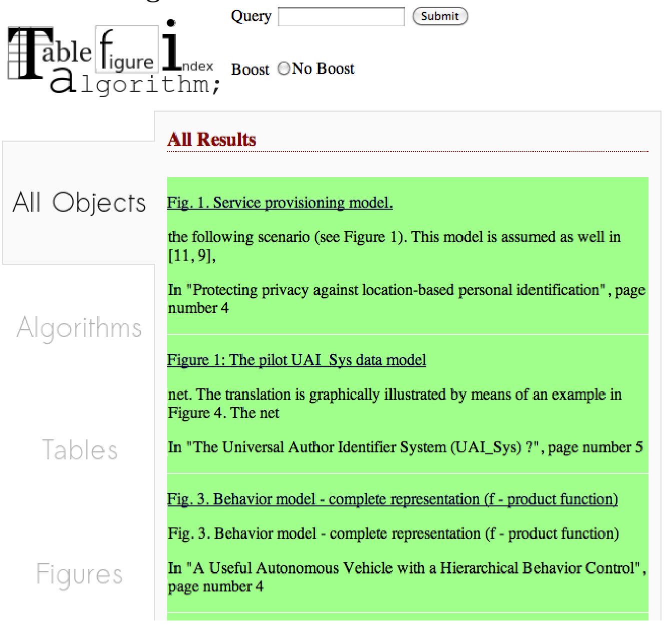


Fig. 1. Service provisioning model.

the following scenario (see Figure 1). This model is assumed as well in [11, 9],

In "Protecting privacy against location-based personal identification", page number 4

Figure 1: The pilot UAI_Sys data model

net. The translation is graphically illustrated by means of an example in Figure 4. The net

In "The Universal Author Identifier System (UAI_Sys) ?", page number 5

Fig. 3. Behavior model - complete representation (f - product function)

Fig. 3. Behavior model - complete representation (f - product function)

In "A Useful Autonomous Vehicle with a Hierarchical Behavior Control", page number 4

choose whether to make use of the query weighing by the application.

4.1 Evaluation

A set of 32 queries consisting of common computer science keywords and phrases were utilized to evaluate the retrieval system built with 5076 objects (1274 algorithms, 1746 tables, 2056 figures). The distributed cumulative gain metric was used for each query, with a resulting combination. Weighing maintained or improved the quality of results obtained for 90.32% of the queries with significant improvement for 19.35% of queries.

5. CONCLUSIONS

We briefly discussed the implementation of a system for indexing multiple embedded object types within a single index. We adopt a common schema, overlap detection and query weighing mechanisms to satisfy user information needs.

The implementation discussed in this article provides a good starting point for further research and development. We hope that this inspires a more detailed and extensive effort to indexing and retrieving diverse objects and an implementation as a stand alone system or part of other digital library systems such as SeerSuite.

6. ACKNOWLEDGMENTS

We would like to acknowledge Anurag Acharya and Pucktada Treeratpituk for useful suggestions.

7. REFERENCES

[1] S. Bhatia, P. Mitra, and C. L. Giles. Finding algorithms in scientific articles. In *WWW*, pages 1061–1062, 2010.
[2] R. Bodner and F. Song. Knowledge-based approaches to query expansion in information retrieval. *Advances in Artifical Intelligence*, pages 146–158, 1996.
[3] H. Cui, J. Wen, J. Nie, and W. Ma. Probabilistic query expansion using query logs. In *WWW*, pages 325–332. ACM, 2002.
[4] Y. Liu, K. Bai, P. Mitra, and C. Giles. Tableseer: automatic table metadata extraction and searching in digital libraries. In *JCDL*, pages 91–100. ACM, 2007.
[5] R. Santos and I. Ounis. Diversifying for multiple information needs. In *ECIR, DDR Workshop*, pages 37–41, 2011.
[6] P. Teregowda, I. Councill, R. Fernández, M. Khabsa, S. Zheng, and C. Giles. Seersuite: Developing a scalable and reliable application framework for building digital libraries by crawling the web. In *USENIX WebApps*. USENIX Association, 2010.
[7] C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *ACM SIGIR*, pages 10–17. ACM, 2003.