MIND: An architecture for multimedia information retrieval in federated digital libraries

Henrik Nottelmann, Norbert Fuhr
Department of Computer Science
University of Dortmund, Germany
nottelmann,fuhr@ls6.cs.uni-dortmund.de

1 Introduction

Today, people have routine access to a huge number of heterogeneous and distributed digital libraries. To satisfy an information need, relevant libraries have to be selected, the information need has to be reformulated for every library w. r. t. its schema and query syntax, and the results have to be fused. This is an ineffective manual task for which accurate tools are desirable.

MIND (which we are currently developing in an EU project) is an end-to-end solution for federated digital libraries which covers all these issues. We start from information retrieval approaches which focus on retrieval quality, but mostly only consider monomedial and homogeneous sources. We will extend these approaches for dealing with different kinds of media (text, facts, images and transcripts of speech recognition) as well as handling heterogeneous libraries (e.g., with different schemas). Another innovation is that MIND also considers non-co-operating libraries which only provide the query interface.

2 The MIND architecture

Our architecture follows the standard structure with one mediator (called "dispatcher" in MIND) and wrappers ("proxies") for every library. The proxies extend the functionality of the non-co-operating libraries and give the dispatcher the required information not provided by the libraries. The extension is implemented with standard implementations and internal textual "resource descriptions" for library-specific information. Despite the communication with the library, only one single proxy implementation (which use different resource descriptions) has to be written. Nevertheless, this approach is flexible enough to allow for different proxy implementations as well.

Thus, for the dispatcher this is the same scenario as if there are only co-operating libraries. The resulting architecture is depicted in figure 1.

The main functionality is shifted into the "smart" proxies. Major tasks are

- query transformation, where a query is rewritten w. r. t. the library schema,
- calculation of expected retrieval costs for resource selection, and
- normalisation of document weights for data fusion.

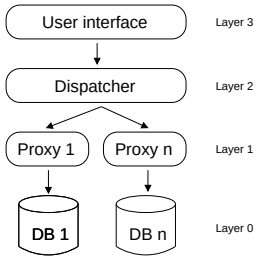

Figure 1: MIND architecture

2.1 Query transformation

A user query is a set of conditions (*weight*, *attribute*, *predicate*, *value*), where the attributes belong to a standard schema. A schema is a set of named and typed attributes [5]. The data type of an attribute defines the media type, the domain and the (vague) predicates.

As MIND considers heterogeneous libraries with different schemas, one major task is to transform the user query into its library-specific counterpart ("proprietary query"). We will use probabilistic Datalog [6] for describing uncertain mappings between schemas.

E.g., a user query condition for the Dublin Core (DC) [3] attribute "creator" will be transformed into three proprietary query conditions referring to the MARC 21 [7] attributes "field 100", "field 700" and "field 710", respectively.

The inverse mapping from MARC 21 as standard schema onto the library schema DC is imprecise, as "creator" is a superset of "field 100" (and retrieval for "creator" returns items which are no solutions for "field 100"). Thus, deterministic predicate logic is no longer sufficient to express the schema mapping. Instead, probabilistic Datalog is used instead, where ground fact and rules have attached a probability instead of boolean weights. This allows us to express the uncertainty of the schema mapping:

$$0.4\ marc\_100(D,V) \leftarrow dc\_creator(D,V).$$

This rule states that 40% of the $(document, value)$ pairs in $dc\_creator$ are also in $marc\_100$. We want the system to learn these rules and conditional probabilities (see [8] for early results).

In general, there can be multiple rules for a head predicate, leading to exponentially (in the number of rules) many probabilities. They have to be coded in the rule bodies (see [8] for a detailed description).

The proprietary query condition weight is calculated based on the user query condition weight and the conditional mapping probabilities.

For text, some (but not all) libraries support stemming. For images, different colour histogram dimensionalities can be used. Thus, mappings have to consider $(attribute, predicate)$ pairs.

The schema definition and the mappings between the standard and the library schema are stored in the resource description (in XML format). Thus, a simple algorithm can automatically transform user queries.

2.2 Resource selection

Resource selection aims at determining a subset of the resources so that both retrieval quality and retrieval costs are optimised when only querying these resources. We use and extend the decision-theoretic model from [4]. Each proxy is asked about its expected retrieval costs (including connection time, computation costs, charges for delivery as well as costs for viewing relevant/non-relevant documents as a measure of retrieval quality). Then, the dispatcher calculates an optimum selection w. r. t. expected overall costs.

Among other things, $E(rel|q, DL)$, the expected number of relevant documents (w. r. t. the query q) in a library DL, is required. As usual in modern IR, we view retrieval as uncertain inference:

$$Pr(rel|q,d) \quad = \quad Pr(rel|q \leftarrow d)Pr(q \leftarrow d) + Pr(rel|\neg(q \leftarrow d))Pr(\neg(q \leftarrow d))$$

$$E(rel|q, DL) \quad \approx \quad Pr(rel|q \leftarrow d) \sum_{d \in DL} Pr(q \leftarrow d)$$

For the linear retrieval function

$$Pr(q \leftarrow d) = \sum_{c_i \in q} Pr(q \leftarrow c_i) Pr(c_i \leftarrow d).$$

where $Pr(q \leftarrow c_i)$ is the indexing weight of condition c_i, this leads to

$$E(rel|q, DL) \approx Pr(rel|q \leftarrow d) \sum_{c_i \in q} Pr(q \leftarrow c_i) \sum_{d \in DL} Pr(c_i \leftarrow d).$$

For every c_i, the proxy has to compute the last sum (of indexing weights).

For text, normalised $tf\ idf$ values can be used as indexing weights. Thus, the sum must be computed only once for every text term.

In contrast, facts and images are represented as feature vectors $\vec{v} \in V$ (e.g., colour histograms) over a continuous domain V. It is not possible to store the sums for all feature vectors. Instead, vectors are clustered. Each cluster $V_j \subseteq V$ is described by its centroid $\vec{v_j}$, the radius and the number of vectors $|V_i|$ in it. Furthermore, let $f : V \times V \rightarrow [0, 1]$ define a retrieval metric for feature vectors. Then, the indexing weight sum can be estimated by

$$\sum_j |V_j|\ f(v_j, \mathit{value}(c_i))$$

for condition c_i (computed at runtime).

All library-specific information needed to compute the expected costs (except the function f which will be coded in the proxy) is stored in the resource descriptions.

2.3 Data fusion

In data fusion, the weights of the retrieved documents have to be normalised for optimum retrieval quality. Several techniques will be investigated. Some of them use global idf values (see [2] for some data fusion techniques).

These global idf values are computed by the dispatcher. For this, it requests local document frequencies (df) for all query terms (and similar frequencies for other media types) from the proxies. These frequencies are also stored in the resource description.

3 Acquisition of resource descriptions

The proxies cannot simply request the resource description from the non-co-operating libraries. For an environment in which the libraries only provide the query interface, query-based sampling has been proposed in [1] as a solution to estimate document frequencies and indexing weights in text libraries. We will extend this technique for other media types, where feature vectors have to be extracted and clustered.

For query transformation, schema mappings are required. We want MIND to learn schema mappings from example sets. The results of early experiments are promising [8], but further work is necessary. Our project will also investigate if the library schema itself can be learned from a sufficient training sample.

4 Conclusion and outlook

This paper describes the architecture of our ongoing project MIND. The fact that the digital libraries are non-co-operating is hidden by proxies which are front-ends for the actual libraries. Thus, for the dispatcher all libraries are fully co-operating.

As all "intelligence" is delegated to the proxies, the focus of our future work will be on the proxies and particularly on the definition, acquisition and use of textual resource descriptions. These contain library-specific information (document/vector frequencies, sums of indexing weights for terms) which can be used by standard implementations of proxies.

5 Acknowledgements

This work is supported by the EU commission under grant IST-2000-26061.

References

[1] J. Callan, M. Connell, and A. Du. Automatic discovery of language models for text databases. In *Proceedings of the 1999 ACM SIGMOD. International Conference on Management of Data*, pages 479–490, 1999.

[2] J. Callan, Z. Lu, and W. Croft. Searching distributed collections with inference networks. In E. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, New York, 1995. ACM.

[3] Dublin Core Metadata Initiative. Dublin Core metadata element set, version 1.1. `http://dublincore.org`.

[4] N. Fuhr. A decision-theoretic approach to database selection in networked ir. *ACM Transactions on Information Systems*, 17(3):229–249, 1999.

[5] N. Fuhr. Towards data abstraction in networked information retrieval systems. *Information Processing and Management*, 35(2):101–119, 1999.

[6] N. Fuhr. Probabilistic datalog: Implementing logical information retrieval for advanced applications. *Journal of the American Society for Information Science*, 51(2):95–110, 2000.

[7] Library of Congress. MARC standards. `http://www.loc.gov/marc`.

[8] H. Nottelmann and N. Fuhr. Learning probabilistic datalog rules for information classification and transformation. In *Proceedings CIKM 2001 (to appear)*, 2001.