LETTER

Predicting DataSpace Retrieval Using Probabilistic Hidden Information

Gile Narcisse FANZOU TCHUISSANG[†a)], ***Student Member***, **Ning WANG**[†b)], **Nathalie Cindy KUICHEU**[†c)], **Francois SIEWE**[††d)], **De XU**[†e)], ***Nonmembers***, ***and*** **Shuoyan LIU**[†f)], ***Member***

SUMMARY This paper discusses the issues involved in the design of a complete information retrieval system for DataSpace based on user relevance probabilistic schemes. First, Information Hidden Model (IHM) is constructed taking into account the users' perception of similarity between documents. The system accumulates feedback from the users and employs it to construct user oriented clusters. IHM allows integrating uncertainty over multiple, interdependent classifications and collectively determines the most likely global assignment. Second, Three different learning strategies are proposed, namely query-related UHH, UHB and UHS (User Hidden Habit, User Hidden Background, and User Hidden keyword Semantics) to closely represent the user mind. Finally, the probability ranking principle shows that optimum retrieval quality can be achieved under certain assumptions. An optimization algorithm to improve the effectiveness of the probabilistic process is developed. We first predict the data sources where the query results could be found. Therefor, compared with existing approaches, our precision of retrieval is better and do not depend on the size and the DataSpace heterogeneity.

key words: *information retrieval, probabilistic algorithm, DataSpace*

1. Introduction

An information retrieval system is characterized by a collection of documents and users who perform queries on the collection to fulfill their information needs. The demand for managing multiple data sources with different data models is rapidly expanding, therefore the need for intelligent management systems providing access to those heterogeneous and often distributed data sources allowing searching and querying them as a single information source, has never been greater; this research challenge is faced by the DataSpaces.

The widely use of information seeking has been greatly expanded. L. Yukun et al. (2009) [1] represents one of the latest efforts on studying issues of context in IR. Similar to ours, users' previous interests and behavior are treated as important clues in those studies; an ontology based on user behavior features is proposed. OrientSpace [1] imported documents and emails from hard disk, email server and web bookmarks using task-centric features, and thus association information between files was discovered. A. Ricardo et al. (2009) [2] suggest links to a user, based on the user's ratings of Web pages. (Mayssam and his colleagues, 2009) [3] combines schema matching and structure discovery techniques to find approximate foreign-key joins across heterogeneous databases. Their learning model is similar in structure to that described in this paper.

EASE [4] models data as graphs, summarizes the graphs, and constructs graph indices instead of using traditional inverted indices for effective keyword search. KRM, the Keyword Relationship Matrix summarized the relationships between keywords in a relational database based on its structure. Data sources are then ranked by comparing the keyword queries with their summaries. iMeMex [5] proposed to unify computer files; it provides an additional persistence and analysis layer on top of the file system. Google Desktop [6] only indexes 100,000 files per drive during the initial indexing period including web browsing history, office documents, instant messenger, and several multimedia file. Our previous work [7], compose information from many data sources to get a best-effort answer and build incrementally semantic relationships in a DataSpace.

Unfortunately, most of those systems are based on the computer files (data sources information), the user query still need to be analyzed (user mind and relevance for example) and they do not formally describe how heterogeneous data sources could be managed in order to predict which sources are suitable for the query.

Our main contributions are twofold:

1- In order to predict the data sources where the query results could be found, Information Hidden Model (IHM) is constructed taking into account the users' perception of similarity between documents. IHM use a segmented query sequence list and a set of heterogeneous data sources, then compute the most likely path to retrieve the efficient result.

2- We introduce the notion of User Hidden Information: UHH, UHB and UHS which can evolve as the system works. The User Hidden Information could express the user mind relatively better so as to increase the precision of retrieval.

The paper is organized as follows: Sect. 2 provides an analysis of DataSpaces Retrieval, Sect. 3 deals with the probabilistic ranking principle, sect. 4 presents the experimental results and sect. 5 concludes.

Manuscript received January 25, 2010.

[†]The authors are with the Institute of Computer and Engineering, Beijing Jiaotong University, China.

[††]The author is with the Software Technology Research Laboratory, Faculty of Technology, De Montfort University, United Kingdom.

a) E-mail: fanzounar2002@yahoo.fr
b) E-mail: nwang@bjtu.edu.cn
c) E-mail: nathkuicheu@yahoo.fr
d) E-mail: fsiewe@dmu.ac.uk
e) E-mail: dxu@bjtu.edu.cn
f) E-mail: 06112062@bjtu.edu.cn

DOI: 10.1587/transinf.E93.D.1991

Copyright © 2010 The Institute of Electronics, Information and Communication Engineers

2. DataSpace

A DataSpace is a virtual space managing many data sources without considering their structure and location. Data sources are then heterogeneous and are considered as "participants"; all kind of "relationships" should be modeled between these participants. A semantic model for DataSpace is:

DataSpace = [***id***: *type,* **Participants**: *type,* **Relationship:** *type*], where **Participant** = [***id***: *type,* ***data-sources***: *type,* ***Description***: *type*, ***Localization***: *type*], **Relationship** = [***id***: *type,* **Participants**: *type,* ***Nature*****:** *type*] and *type* is the Datatype or the record label.

Participants vary from being very structured to semi-structured to completely unstructured. A *structured* data consists of named components, organized according to some well-defined syntax, e.g. a relational database. A *semi-structured* data has no fix schema, the structure is implicit and irregular, e.g. XML data, web pages. The term *unstructured data* refers to data which does not have clear data model, semantically overt, e.g. a picture, audio, video, e-mail.

2.1 DataSpace Information Retrieval

Querying and searching are two main information retrieval methods supported by a DSSP (*DataSpace Support Platform*). A well known search is the keyword search. Query translation methods needs intelligent methods for interpreting and translating queries into various languages. To resolve the problem of language translation, we proposed in [7] to users to give a query as a keyword or a picture. Users type a query, DataSpace information are analysed to provide a formal representation of their contents. The query is matched against entries in the index in order to determine which information are relevant to the user.

A data source is a set of domains; a domain is a set of information (*can be a collection of documents*). A domain can be represented in the vector space model by a term-domain matrix. T^i is a given term i and w^i_j is the weight of the term T^i over the domain D_k of the data source DS_j,$tf_{ij} = f_{ij} / \max_i\{f_{ij}\}$ is the term frequency of term T^i in the domain D_k, Df_i *is the domain frequency and* k *is the number of domains.* The Retrieval Status Value (*RSV*) is the function or a probability of similarity between a domain and the query, using Bayes' theorem, $\mathbf{R}SV(\mathbf{q},\mathbf{D}) = \frac{\mathbf{P}(\alpha_i)\mathbf{P}(\mathbf{w^i_1},\mathbf{w^i_2},\ldots,\mathbf{w^i_n}/\alpha_i)}{\mathbf{P}(\mathbf{w^i_1},\mathbf{w^i_2},\ldots,\mathbf{w^i_n})}\alpha_i$ is the weight of term T^i in the query q and w^i_j is the weight of term T^i in the domain D.

3. Probability Ranking Principle by Example

3.1 Elementary Sub Queries (ESQ)

A query is decomposed into ESQs and since the DataSpace is composed of Data Source (DS) the system would used the RSV to find which ESQ can be found in Which DS. To achieve this task, our system uses both user query and DataSpace hidden information, in fact we believe that the user query is hiding the user mind, hence we can find those user hidden needs by taking into account three different learning strategies:

- **User Hidden Habit** (UHH): *for the ten first queries of the user, the system will analyze the results that interest the user, taking into account the results selected by the user; for more queries, UHH evolves as the system works.*

- **User Hiddden Background** (UHB): *after analyzing the results selected by the user, our system will classify the user according to his speciality. E.g if a user selects many results related to computer, the user UHB will be "computer specialist" and next time if the user type "system" as query our system will first return results around computer such as "computer system, network system, database system. . . "*

- **User Keyword Hidden Semantics** (UHS): *using an ontology dictionary, our system will redefined user keywords to find synonym and new sub keywords related to the user query.*

UHH, UHB and UHS descriptors produced a set of Elementary Sub Queries which will be used to represent the query with the aim of retrieving the user mind from the user query. Corresponding DS are requested using ESQs in order to retrieve the best effort result.

3.2 Information Hidden Model (IHM)

IHM allow to estimate probabilities of unobserved events of a Data Source, e.g., in speech recognition, the observed data is the acoustic signal and the words are the hidden parameters. IHM is a kind of information about information, a semantic description of data with probabilistic generative model for sequences.

To illustrate the feasibility of our system, we will consider a user querying a Medical DataSpace of 3 participants. An IHM is a rewriting system denoted as $(\mathbf{D}SP, Q, \mathrm{O})$: *i) A set of Data sources* $DSP = \mathbf{\{DS_1, DS_2, \ldots, DS_N\}}$. *ii) A decomposed set (ESQ) of user query* $Q = \mathbf{\{q_1, q_2, \ldots, q_M\}}$; $\forall i \neq j,\ q_i \cap q_j = \emptyset.$; *iii) Output result sequence (Observations):* $O = \mathbf{\{O_1, O_2, \ldots, O_M\}}$.

For example, if N=3 and M=4 then DSp = {DS_1, DS_2, DS_3} and Q ={q_1, q_2, q_3, q_4}. The first purpose of IHM is to find O, hence $\mathbf{\{O_1, O_2, O_3, O_4\}}$**?** see Fig. 1 for hidden probabilities and observations.

$\boldsymbol{a}_{ij}$ is the transition probability from DS_i to DS_j *(if there is any relationship between* DS_i and DS_j*)*; $\boldsymbol{b}_{it}$ is the probability of observation o_t being emitted by DS_i. Note that $\sum_j^n a_{ij} = 1$, $\sum_k^m b_{ik} = 1$, $\sum_k^m \sum_j^n a_{ij} b_{ik} = 1$. We can generalised our model as:

$IHM : (DSP, \mathbb{Q}) \rightarrow O; (DS_i\ ,\ q_i\ ) \rightarrow \mathbf{IHM}_i$.

Each data source state (Fig. 1) contains 2 descriptors: *the retrieval status* $\mathrm{RSV}(q_i, DS_i)$ and the possible *hidden information IHM*(DS_i, q_i). O_t is the data source observator at time t. We accept observation O_t =IHM(DS_t, q_t) when

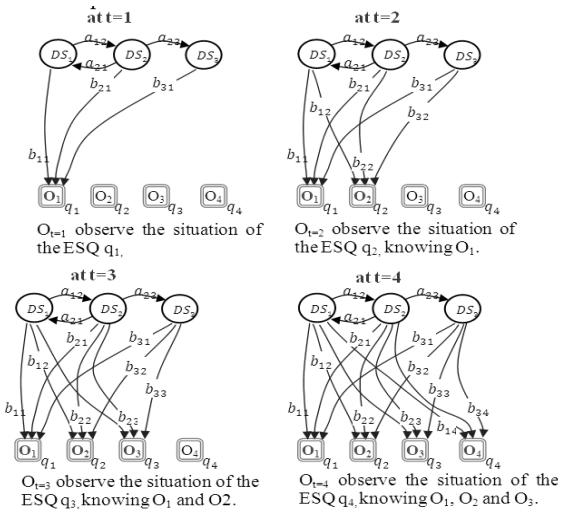


Fig. 1 IHM retrieval principle.

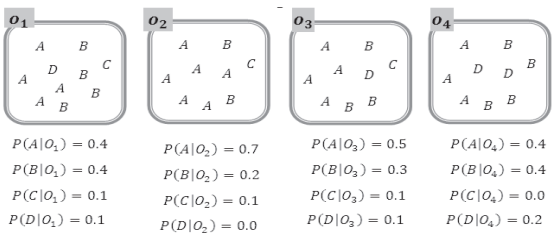
$RSV(q_t, DS_t) \geq 0.4$, then the hidden information becomes the ESQ result at this observation state at time t. Now that we have results of ESQs (O=**{O$_1$, O$_2$, O$_3$, O$_4$}**), the second step is to combined those sub results to retrieve the final result, see Fig. 2.

We then faced the combining problem:"*Given a DataSpace DSP = DS_1, DS_2, …, DS_N, Find the result sequence $O = O_1, O_2, \ldots O_T$ of the query Q=$q_1, q_2, \ldots, q_m$such that $P(Q/DSP)$is maximized. We want to find the path with the highest probability*". Using this probability, the system can easily retrieve the efficient result for the user query. We used the dynamic programming to solve this problem, using Bayes rule $P(Q/DSP) = \frac{P(DSP/Q)*P(Q)}{P(DSP)}$, $P(DSP) = C$ is constant since it did not depend on the query, hence

$$\mathbf{P}(\mathbf{Q}/\mathbf{D}SP) = \frac{1}{C}P(Q)\prod_{j=1}^{n}\mathbf{P}\left(\mathbf{D}S_{\mathbf{j}}/\mathbf{q_j}\right) \quad (1)$$

From Eq. (1), to maximized $P(Q/DSP)$, we simply need to maximize each $P\left(DS_j/q_j\right)$ for all j.

Let O={O$_1$, O$_2$, O$_3$, O$_4$} be the sub result set of the first step, the hidden discrete observation probabilities are:



Hence R= {A, B, C, D} is the result set of the observatory O = {o$_1$, o$_2$, o$_3$, o$_4$}. *Note that A, B, C, D= IHM*(DS$_t$, q$_t$), are the hidden information at time t.
P(A|O)=(0.4+0.7+0.5+0.4)/4=0.5,
P(B|O)=0.325, P(C|O)=0.075, P(D|O)=0.1,

```
Algorithm Combine (guessState, sequenceIndex):
1  if sequenceIndex is past the end of the sequence then return 1
2  if (guessState, sequenceIndex) has been seen before then return saved result
3  else result = 0
       for each neighboring state n:
              result = result + (transition probability from guessState
              to n given observation element at sequenceIndex)
              * Combine (n, sequenceIndex+1)
4     save result for (guessState, sequenceIndex)
5     return result
Where guessState is a new Data source, sequenceIndex is the set of data sources
already indexed (already combined).
```

Fig. 2 IHM combining algorithm.

$$P(RjO) = \sum_{i=1}^{4} P(O, R_t = i) = 1$$

The result of the query Q can be found in two domains set: A and B since $P(A \mid O)$ *and* $P(B \mid O)$ are those maximizing **P (R | O). The efficient data sources are those predicting A and B.**

Assumption: The IHM assumption states that probability of the occurrence of query result R$_i$ at time t depends only on occurrence of result R$_{i-1}$ at time t-1.

4. Experimental Results

We implement the proposed IHM algorithm using C++Builder 6.0 and PHP 5.0 on a mixed network with 3 computers, one on windows Xp, one on Linux-Fedora 10, and one on Ubuntu Desktop 9. Each computer using 2 CPUs Intel Pentium M 3 GHz with 1 Gb memory. We compared IHM performance wit Google Desktop v5.9, OrientSpace v0.9 and iMeMex-src-0.46.2 performances. We measured **i)** *the precision of retrieval (percentage of similarity between the query and the result (PR))*. **ii)** *Time of results*. It has been subjected to heavy testing in the very large evaluation program represented by the TREC [8], Text REtrieval Conferences; using the TREC dataset and the basic processing operations of search: exact match, semantic and full text search; see Fig. 3.

- Exact match search (see Fig. 3 a): for 100 IHM retrievals, about 97 are relevant to the user, because IHM used the user mind and all returning results are selected by the user; the UHH (User Hidden Habit) descriptor accumulate the user habit and used it to retrieve more relevant result. For more than 250 retrievals, Google Desktop returned many non relevant results, i.e. 40 are not selected by the user; due to the limited amount of indexation on Google Desktop (because only 100 000 files are indexed). 24.61% of retrievals are not similar for OrientSpace and iMeMex.

- Semantic search (See Fig. 3 b): for less than 250 retrieved results, all the systems are dealing with the same precision of retrieval, for more than 250 OrientSpace PR is 1.5% more than IHM and 3.5% more than others, due to the association information between files and user tasks created by OrientSpace. IHM is closed to OrientSpace because of the multiple annotations and ontology used by UHS (User Keyword Hidden Semantics).

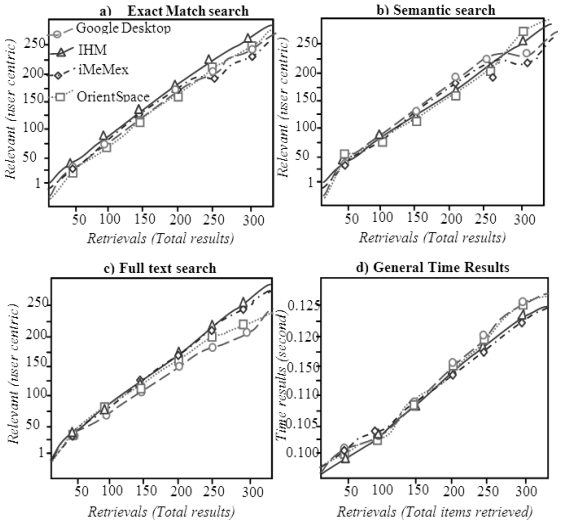


Fig. 3 Precision of retrieval and time results.

- Full text search (See Fig. 3 c): the PR of iMeMex and IHM is closed to 78.4% while that of Google Desktop and OrientSpace is 74.35% for less than 250 retrievals, and IHM PR is 71.45% for more than 250, due to the information clustering done by UHB (User Hidden Background) descriptor in IHM and the files unification in iMeMex.

- Time results (See Fig. 3 d): The chart shows that our four search systems are running at the same response time, as for retrievals under 250. For more than 250 retrievals, iMeMex is a little bit fast (0.144 seconds) than IHM (0.148 seconds). For more than 300 retrievals IHM and OrientSpace are faster than others. For more than 300 retrievals Google Desktop is running at the same response time as iMeMex.

In general, IHM is constant on chart a, b, c and d; from 50 to 300 retrievals, the IHM chart is growing constantly, and the chart is almost a line, this proved that IHM is running in a good precision of retrieval and IHM do not depend on the size and the heterogeneity of the DataSpace.

5. Conclusion and Futher Work

This paper presents a predictive DataSpace retrieval system, based on user relevance probabilistic schemes. Information Hidden Model (IHM) is constructed taking into account the users' perception of similarity between documents in order to well represent the user mind, and to describe a data source according to a given query. The goal of such approaches is to define flexible Information Retrieval Systems able to deal with the inherent vagueness and uncertainty of the retrieval process. The system first accumulates feedback from the users and employs it to construct user oriented clusters, and then predict the data sources where the query results could be found. In order to generalize our technique, we will demonstrate how a user adaptive technique can be used to retrieve information for ubiquitous computing environment.

References

[1] Y. Li, X. Meng, and X. Zhang, "Research on dataspace," J. Software, vol.19, no.8, pp.2018–2031, 2008.
[2] A. Ricardo , L. Leydesdorff, C.-R. Zaida, R. Rousseau, and P. Soren, "Retrieval of very large numbers of items in the web of sciences," J. El Profesional de la Informacion, vol.18, no.5, pp.529–533, 2009.
[3] S. Mayssam, H. LeKhac, A. Doan, and L. Gravano, "Efficient keyword search across heterogeneous relational databases," 23rd IEEE Int. Conf. on Data Engineering (ICDE), vol.15, no.20, pp.346–355, 2007.
[4] L. Guoliang, F. Jianhua, W. Jianyong, and L. Zhou, "EASE: An effective 3-in-1 keyword search method for unstructured, semi-structured and structured data," ACM-SIGMOD Int. Conf. on Management of data, pp.903–914, Auckland, New Zealand, 2008.
[5] J.-P. Dittrich, "iMeMex: A dataspace odyssey: The iMeMex personal dataspace management system," 3rd Biennial Conf. on Innovative Data Systems Research (CIDR), California, USA, 2007.
[6] B. Jim , D. Kapoor, C. Knoblock, and K. Lerman, "Information integration for the masses," J. Universal Computer Science, Special Issue on Wrapping Web Data Islands, vol.14, no.11, pp.1811–1837, 2008.
[7] C. Kuicheu , N. Wang, G. N. T. Fanzou, F. Siewe, and D. Xu, "Building semantic relationships incrementally in DataSpace," IEEE Int. Conf. on Information Science and Engineering (ICISE), Nanjing, China, Dec. 2009.
[8] TREC, "Experiment and evaluation in information retrieval," http://trec.nist.gov/, July 2009.