Intelligent Information Retrieval in a Digital Library Service

G. Semeraro, F. Abbattista, N. Fanizzi and S. Ferilli
{semeraro,fabio,fanizzi,ferilli}@di.uniba.it
Dipartimento di Informatica
Università di Bari
Via E. Orabona,4 70125 Bari, Italy

Abstract. We present the Private Digital Library (PDL) project that represents a service of the Corporate Digital Library (CDL) prototype. The main ideas underlying this project are the following. When a user is looking for documents that he already retrieved in the past, he has to repeat the search procedure and solve the same problems he encountered in the past access. On the contrary, consider the possibility for the user to store documents in a private library. In this case, he will have the chance to retrieve the documents of interest more easily and quickly. It may also be the case that the user has not a clear understanding of what he is looking for. Moreover, he might not know exactly the library content and organization. Thus, he needs for assistance to suggest him what to look for and how to query the system. A method to store and catalogue documents according to personal criteria might help to overcome these problems.

1 Introduction

In the last ten years, digital libraries became a prominent research area. Main goals of projects involving digital libraries are the application of the several information retrieval techniques developed in the 80's and the realization of new distributed technology to manage information resources.

The typical problem of document retrieval in a digital library can be overcome if users have the possibility to manage their own personal library.

We present the Private Digital Library (PDL) project that represents a service of the Corporate Digital Library (CDL) prototype, developed at the LACAM Laboratory [2], [4], [7]. The aim of this project is to enable users of CDL to generate locally a personal library where to store their preferred documents once they are retrieved from the CDL. Moreover, the PDL system incorporates an assistant agent to help users in formulating their queries, in case they are inexperienced or have not made clear their information need.

2 Private Digital Library

When the user receives satisfactory results of a query (performed through the CDL) he may choose to store all or part of the results in his/her *private library*. On the other hand, if the user is not satisfied by the results of the query, he can select, from the documents found, only some portions of text units, and use them to populate a repository of text units. The documents in the private library are organized by the *Cataloguer* module in proper folders, according to user preferences. The user can decide to find documents contained in his private library, by activating a local *Search Engine*. By means of a certified applet, the whole system can be hosted on a server and the user can access the system through a client.

The user is central in the architecture. When the user starts the interaction by proposing an initial query, he receives the results of the query (documents found in the CDL) he may choose to store all or part of the results in his/her PDL. On the other hand, if the user is not satisfied by the results of the query he can select, from the documents found, only some portions of text units, and use these portions to populate a repository of text units. This repository represents the input to the *Query Suggester*, one of the services offered by the PDL [9]. This agent, on the basis of the text units stored in the repository, helps to formulate queries in a more convenient way, so to retrieve more documents of interest from the CDL to be subsequently inserted in the PDL.

The query suggester module performs the extraction of keywords to be proposed for new queries, exploiting the reduced amount of text units stored in his personal repository that, in a way, represents the user interest. Figure 1 shows the architecture of the query suggester and highlights the course of actions previously described. Once invoked from the user, the query suggester performs the three following phases:

Phase I: Folders and repositories creation. In this phase, the user selects relevant text units among the documents returned by the CDL. The selected text units are stored in the local repository. If the user needs to repeat the document retrieval, he will use the text units of the local repository, in order to speed up the process, without performing again the search in the CDL. This procedure is less computationally expensive than the extraction of keywords from whole documents. The user may update the content of the local repository by adding or deleting text units. Updating the local repository allows for the extraction of more effective keywords and, consequently, the formulation of more refined queries. For each of his

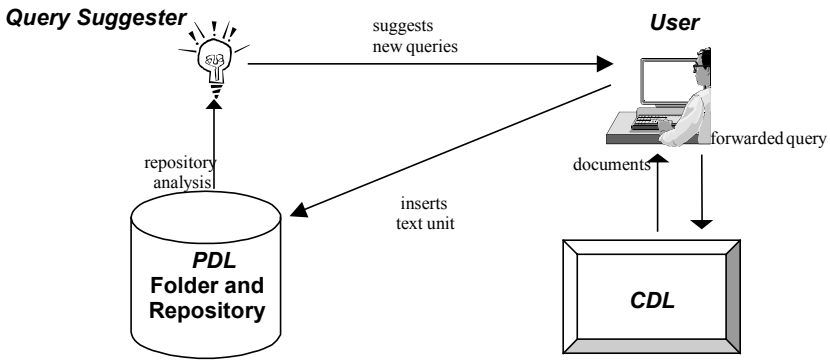


Figure 1. The Query Suggester in PDL.

interests and preferences, the user may create a local folder to better organize the content of his own private library.

Phase II: Pre-processing. The set of text units is transformed in a list of words. This list is reduced by deleting all the words belonging to a list of *stop words* (all the words with low semantic content like adverbs, articles, pronouns, etc.). Porter's algorithm is applied to this reduced list, to delete the suffixes in order to cluster all words with a common root. The tokens obtained will constitute the list of words.

Phase III: Queries formulation. The list of words constructed in phase II is represented as a matrix. Each element of the matrix contains the TFDF's (Term Frequency per Document Frequency) value of i-th words in the j-th text unit. The TFDF technique has been adopted because we think that suggested queries should be made by terms in the repository corresponding to words that are more relevant with respect to a text unit and contained in most of the text units of the repository. The keywords for the new query formulation are extracted according to the following algorithm [9]:

Step 1 For each row of the matrix A(*m x n*), the system evaluates the squared norm and it extracts from the repository the four keywords with the maximum value of the norm.

Step 2 For each of the selected keywords, the system identify more terms, with higher correlation value and these words will constitute one of the suggested query. To compute the correlation value, the system evaluates, for each column of the matrix, the squared norm and, for each term T extracted on step 1:

(a) Define vector V, of size *m*, where for each element *i*, V[i]=0 if the word T does not belong to the i-th text unit, otherwise V[i]=the evaluated squared norm.

(b) Identify the text unit corresponding to the vector with the maximum value of the squared norm.

(c) Extract, from the identified text unit, two words, W1 and W2 with the higher values of TFDF.

(d) Formulate the query composing T, W1 and W2.

The process is iterated to compose 4 queries. Figure 2 shows an example of queries extracted by the algorithm presented above.

The PDL cataloguer enables the user to organize and store in the PDL all documents of interest retrieved from the CDL. The user can create different folders in his own PDL, one for each interest or preference, thus the cataloguer allows for the management of these folders and their content. The user creates a new folder every time he wants to add a new topic in his PDL. The documents stored in the PDL can be displayed on demand. For each document, the system will display the following information: *authors*, *title*, *abstract* and the *body* of the document. The user may also add a comment about a document or move documents among folders.

The PDL search engine allows searching for documents in the PDL. The user may choose among three different search criteria such as: *title*, *author* and *subject*. The user can choose a single criterion or any combination of the three criteria mentioned above. The system displays a list of documents matching the selected criteria and that can be consulted.

3 Experimental Results

Experiments have been carried out in order to test the efficiency of the Query Suggester in the PDL.

The main goal has been to check if the mechanism of query formulation, assisted by an agent, gives a suitable tool for augmenting the number of significant documents, extracted from the CDL, to be stored in the PDL.

Besides, it has been tested also how many steps are necessary for retrieving the most of the important documents for the user, filtering the queries.

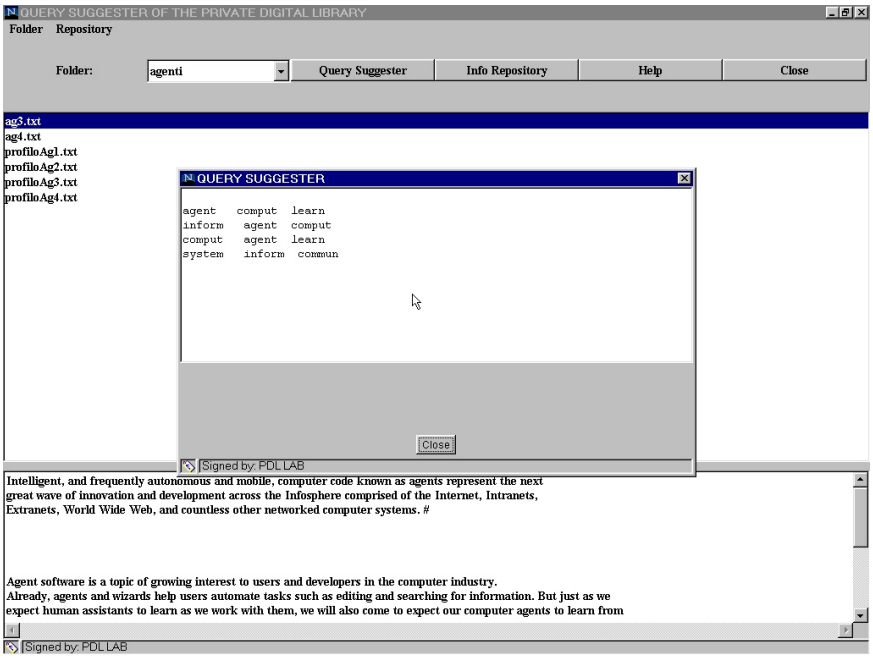


Figure 2. An example of suggested queries.

3.1 Experimentation plan

During the experimentation, heuristics and measures that are commonly adopted in Information Retrieval have been used. They are a sequence of parameters related to the efficiency of such systems [5][6]. Precision and Recall [8] are the two parameters that have allowed measuring the goodness of the queries realized by the Query Suggester. In order to measure these parameters, it has been necessary to store a sample of 100 documents extracted by the CDL (in particular from the library AI_IN_DL, a test digital library containing articles on artificial intelligence). It has been supposed that the user goal was the identification of objects in the images and it has been verified the presence of 20 documents of the sample related to such argument. Furthermore, the search in the user's PDL has been conducted by means of AltaVista Discovery, that is a search engine for personal computer used in order to carry out the suggested queries.

The user begins the search devising the starting query Q. In the example shown in the following let us suppose he/she starts with Q = "*learning*". The outcomes represented in the following table display the number of important documents, the total number of documents retrieved and the values of precision and recall obtained.

Query	Relevant Documents	Retrieved Documents	Precision	Recall
Q	6	62	9,68%	30%

At this point of the search the user analyses the retrieved documents and stores the important text-units in his personal repository from which the Query Suggester will formulate a set of four queries. During the experimentation, eight different repositories have been chosen, each containing an increasing number of text units (1,3,5,7,9,11,13,15) where the larger repository contained the smaller ones. In the following, the first repository, containing a text-unit, is shown.

*Repository with **1** text unit*

Recognize1

We present a new approach for tracking roads from satellite images, and thereby illustrate a general computational strategy ("active testing") for tracking 1D structures and other recognition tasks in computer vision. Our approach is related to recent work in active vision on "where to look next" and motivated by the "divide-and-conquer" strategy of parlour games such as "Twenty Questions." We choose "tests" (matched filters

for short road segments) one at a time in order to remove as much uncertainty as possible about the "true hypothesis" (road position) given the results of the previous tests.

From their analysis new related queries have been extracted. For instance, from the analysis of the first repository shown above, the Query Suggester formulates the following four queries:

Q1: *road test approach* **Q2**: *test road approach*
Q3: *comput test road* **Q4**: *vision test road*

Table 1 shows the results of the search obtained through the four suggested queries.

The relative values of Precision and Recall are represented in Figure 3, showing their average trends, along with the increasing number of text-units in the repository.

3.2 Results and Discussion

From the diagrams in Figure 3, it is possible to notice how, initially, such values are really small (9,68% and 30%). The reason of these results has to be found in the absolute non-specific quality of the starting query, due to the user's lack of a perfect knowledge of the related argument.

However, by the selection and storage of important text units in the repository, the user is able to obtain queries that return more relevant documents. For the queries suggested by a repository containing 9 text units it is possible to obtain: 79,66% value of *Precision* and 75% value of *Recall*.

Besides, analyzing the suggested queries, it is possible to observe that they are the same for repositories containing 11, 13, 15 text units with *Precision* and *Recall* values of 60,18% and 75%. Therefore the adding of further text units is not relevant because the results became stable from repositories containing 11 text units. Before the experimentation, it was thought that the result of the suggested queries would be increased with the growing of the repository dimension. Indeed the two graphs show how the trend is normally increasing, but there are some falls due to the non-importance of the text units introduced.

The conclusion to be drawn is, as expected, that the effectiveness of the suggested queries depends both on the repository dimension and on the significance of the text unit that form the repository.

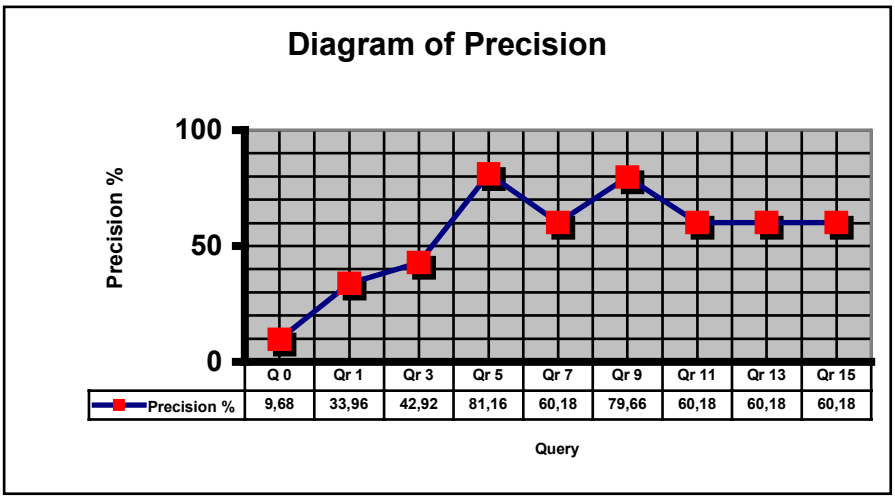
4 Conclusions and Future Work

We presented the Private Digital Library project, one of the services available from the Corporate Digital Library prototype. An intelligent agent was illustrated for assisting the user by suggesting improved ways to query the system on the ground of the documents stored in a personal catalogue according to his own preferences, which come to represent his interests.

Future work will concern the exploitation of information coming from the rejection of some retrieved text units or whole documents so to specialize his model of the user interest and thus further refine the suggested queries.

Query	Relevant Documents	Retrieved Documents	Precision	Recall
Q1	16	67	23,89%	80%
Q2	16	67	23,89%	80%
Q3	7	22	31,82%	35%
Q4	18	32	56,25%	90%
		Mean Q_r1	33,96%	71,25%

Table 1. Results obtained by refining the Query in the example.



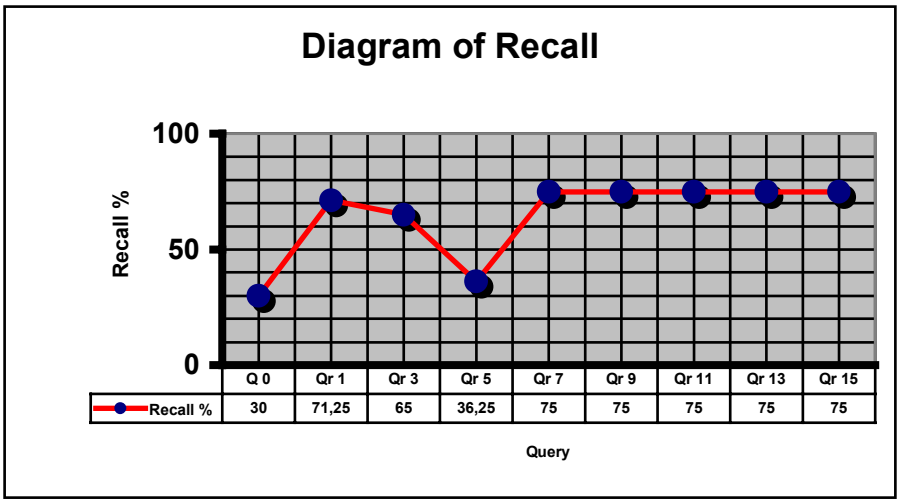


Figure 3. Precision and Recall Diagrams.

Acknowledgements

The authors would like to thank G. M. De Nuccio and M. A. Giuzio for their precious collaboration both to the system implementation and to the experiments performed.

This work was partially funded by the COGITO Project IST-1999-13347.

References

[1] T. Cea, *Interazioni con agenti di ricerca: l'adeguamento alle esigenze informative dell'utente*, Tesi di Laurea, Dipartimento di Informatica, Universita' degli studi di Bari, a.a. 1997-98.

[2] M.F. Costabile, F. Esposito, G. Semeraro, N. Fanizzi, *An Adaptive Visual Environments for Digital Libraries*, Int. J. On Digital Libraries, 2(2/3):124-143, Springer-Verlag, Berlin, 1999.

[3] I. Di Fonzo, *Interazioni con agenti di ricerca: il Sistema Suggy per la formulazione automatica delle query*, Tesi di Laurea, Dipartimento di Informatica, Universita' degli studi di Bari, a.a. 1997-98.

[4] F. Esposito, D. Malerba, G. Semeraro, N. Fanizzi, S. Ferilli, *Adding Machine Learning and Knowledge Intensive Techniques to a Digital Library Service*, Int. J. On Digital Libraries, 2(1):3-19, Springer-Verlag, Berlin, 1998.
[5] D. Gilbert, *Intelligent Agents: The Right Information at the Right Time*, IBM Corporation, 1997.
[6] G. Salton, M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.
[7] G. Semeraro, M.F. Costabile, F. Esposito, N. Fanizzi, S. Ferilli, *A Learning Server for Inducing User Classification Rules in a Digital Library Service*, in Z.W. Ras and A. Skowron (eds.), Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence 1609, 208-216, Springer, Berlin, 1999.
[8] K. Sparck Jones, P. Willet, *Readings in Information Retrieval*, The Morgan Kaufmann Series in Multimedia Information and Systems, Edward Fox, Server Editor, 1997.
[9] F. Abbattista, F. Esposito, N. Fanizzi, S. Ferilli, G. Semeraro, *Suggy: An Automatic Query Refinement Agent*, Proc. of the Workshop on Machine Learning in the New Information Age, European Conf. on Machine Learning, Barcelona, 2000.