FameIr, a Multimedia Information Retrieval Shell

M. Bertran, M. Gatius, H. Rodriguez
TALP Research Center, Universitat Politècnica de Catalunya
mbertran@lsi.upc.es
gatius@lsi.upc.es
horacio@lsi.upc.es

This paper describes FameIR, a software environment developed for providing IR capabilities, both indexing and retrieving, when information is represented in different media (text in various languages, audio, images, video, etc.). FameIR also allows access to information in Web pages and information services.

Keywords: Multimedia Information Retrieval; Multilingual Information Retrieval; Web pages; Web documents

1. INTRODUCTION

Although the initial and, up-to-now most widely-spread, form of Information Retrieval (IR) reduces the search space to closed collections of textual documents and the queries to (structured) lists of terms typed through the keyboard, there is a rising interest in i) going beyond closed collections for facing IR over open and changing information sources as the Internet, and ii) allowing other modalities for both queries and collections in what is commonly referred to as *Multimedia Information Retrieval* (MMIR). Some authors (see [5]), restrict the term MMIR to the case of multimedia collections, regardless of the query mode, and use instead the term *Multimedia Query Processing* (or frequently *Spoken Query Processing*, SQP) when it is the query that occurs in non-textual form.

Different media can be used to support the content of the collections of documents (text, audio, video, images, etc.). Sometimes, individual documents are themselves multimedia (e.g. textual documents including images, video documents including voice or textual captions, multimedia Web pages, etc.).

Besides, the fast growth of Internet has made available a large amount of information that cannot directly be accessed by conventional IR technologies, especially that information generated by web services based on user request.

This paper describes *FameIR*, a software environment designed and built to provide some of these extended IR capabilities within the framework of the *FAME*[1] project.

2. MULTIMEDIA DOCUMENTS

The most extended way of dealing with multimedia documents (or multimedia collections) is attaching a set of textual features to each document during the indexing phase. Indexes are then built from these sets of textual features and the query phase is basically the same as in conventional IR. Queries in this case consist of textual features and the matching process is carried out between textual features. As a result of the searching process, the information offered to the user consists of textual information and only in the case the user wants to access a retrieved document a specific process depending on the media of the document is carried out.

Textual features can be words, stems, lemmas, multiword terms, phrases, etc. They can include several types of information: morphological (e.g. part of speech), syntactic (e.g. syntactic category) and semantic (e.g. sense). Textual features can be automatically extracted from multimedia documents or manually attached to them.

In the case of speech documents (or video documents including voice) two alternative options can be chosen. Some systems use *Automatic Speech Recognition* (ASR) modules for automatically translating speech to text and then proceed as in the previous case. As expected, this ASR process is not error free and thus the accuracy of the indexing process suffers some drop. The other alternative consists in manually transcribing these documents and then indexing the resulting transcripts as described above. This process

assures a higher accuracy at a higher cost of human effort.

Documents containing images or video without voice or text explanations are more difficult to index automatically. In most cases, indexing is performed manually and even in the case of carrying out an automatic scene analysis, human intervention at a post-process step is usually needed.

A remarkable example of this kind of approach is ATT's SCAN (Spoken Content-based Audio Navigator) system ([4], [12]). The system allows indexing and querying of broadcast news stories. Indexing starts by segmenting each document by topic, detecting paratones (intonational paragraph boundaries) using decision trees techniques. Next, a standard ASR is performed (a WER of 32.4% is reported by the authors) followed by conventional indexing of recognised words using SMART. Querying is performed through the SCAN user interface (SCAN UI) that allows relevance judgements, fact finding and summarisation among other query facilities.

A well-known problem in IR is the *Term Mismatch* problem. This problem occurs when a relevant document is not retrieved due a term in the query does not occur in the document. In the case of spoken documents (resulting from ASR) the problem is worsen because of the *Term Misrecognition* problem. This problem occurs when a relevant document containing all the terms of the query is not retrieved because some of the terms have been misrecognized by the ASR.

Several authors ([11], [5], [6], [8]) have proposed different approaches to face both the *Term Mismatch* and the *Term Misrecognition* problems. Among these proposals are document clustering, dimensionality reduction, document and query expansion, semantic or phonetic similarity between words as an alternative to usual exact-match measures, etc.

In [9], [10] and [14] we can find excellent reviews of the problems and techniques presented in this paper.

3. USING FAMEIR

FameIR can manage several collections of documents referred as MULTIMEDIA collections. These collections are sets of documents that can contain multimedia data. Each document, depending on its type, can be associated with files containing complementary material for addressing management of multimedia data. Considering the different media used to support the content of the document there can be five kinds of documents:

- Text document.
- Image document. It consists of a file containing the image and, optionally, an associated file with a description of such image (as a bag of keywords). If the description is not available, the system generates a (minimal) default one.
- Voice document. It consists of a voice file that can be associated with a transcription file (generated from the voice document by an ASR module or manually transcribed or revised) and a file containing the keywords extracted from the transcription.
- Video document. It consists of the video files, the transcription file (as in the video document) and a keyword file.
- Web document. It consists of the keyword file describing the web source and a file containing the following information: the URL of the web document, the description of its structure and, in case of web services, the parameters needed for accessing it.

There are several scenarios where FameIR capabilities are used within FAME environment. Two examples of them are briefly described next. In the first situation, a set of related collections (a recorded meeting and material referred during the meeting) is accessed using conventional IR techniques. In the second, the collection of documents is accessed online using the terms occurring in the conversation.

Figure 1 shows a schema of the first scenario. A meeting has taken place and has been recorded. The whole recorded meeting has been stored as a single MULTIMEDIA collection. The documents of this collection are voice files corresponding to participant's turns within the meeting. Each document has attached several metadata, such as time tag, participant identifier, confidence scoring of the ASR recognized units, etc. In this case, some pre-processing is needed. The voice files have been translated into **.asm** files by the ASR module (although a manual transcription could be done instead). Additional material could also be indexed and retrieved during the meeting.

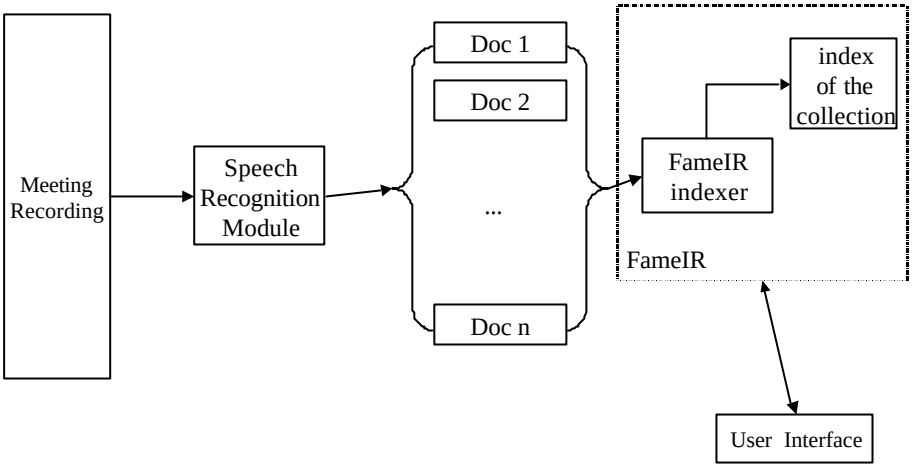

Figure 1. Conventional IR access to a Recorded Meeting.

In this scenario the query phase is performed conventionally. The documents consulted are basically those in the collection representing the meeting although other documents previouly indexed, can also be accessed. Users can type queries like the following:

Who was the first one to talk about Gaudi?

(The system will retrieve the document corresponding to the turn of the first participant in the meeting talking about *Gaudi*)

Gaudi AND "La Pedrera"

(The system will retrieve all the interventions indexed by the terms *Gaudi* and *"La Pedrera"*)

When was modernism introduced?

(The system will retrieve the intervention of the first participant talking about *modernism*)

In this situation, the difficulties are restricted to the indexing task. The documents are retrieved (together with their confidence scores) from the collection by the Speech Recognition Module (SRM), as seen in figure 1. These documents consist of hypotheses about what has been said. When weighting the terms of the documents, not only conventional *tf* (term frequency) and *df* (document frequency) is considered, but also the confidence scores of the hypothesised terms. Regarding querying, several forms of query expansion are presented to the user.

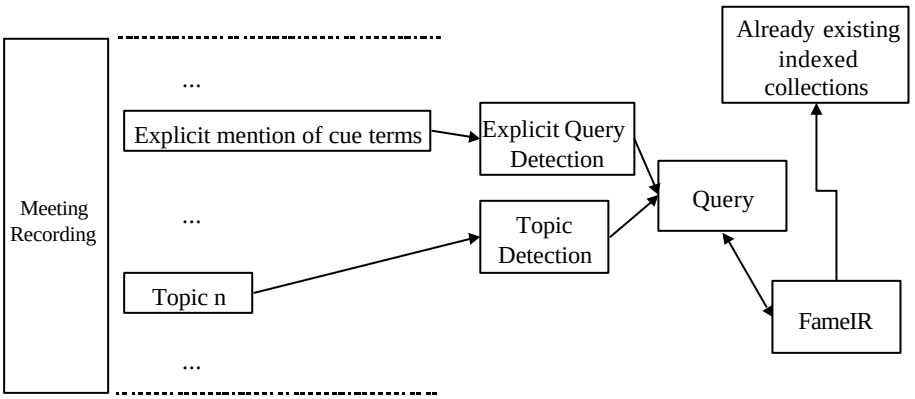

Figure 2. Implicit and Explicit triggering of IR

Figure 2 shows a schema of the second scenario. In this situation, a meeting is taking place and relevant documents are accessed online from a set of collections triggered by explicit or implicit mention of relevant terms during the conversation. A set of collections of related material (basically texts and images) has been previously collected and indexed. Some of the material comes from Web pages. During the conversation, explicit IR queries can be performed such as: *Can we get a picture of "La Sagrada Familia"?*. Otherwise, the system can detect some topics associated with relevant query terms that can act as implicit queries, as in: … *when Gaudi started the building of "La Sagrada Familia"*…

In this second scenario, all the material has been collected, pre-processed and indexed in advance. As already said, two kinds of queries can be considered, explicit and implicit. The former requires an *Explicit Query Detection Module*. The latter requires a *Topic Detection Module* to convert its current formulation into a real query, as seen in figure 2. In this case, new difficulties arise in the detection and extraction of the queries terms from the sentences.

4. FAMEIR FUNCTIONALITIES AND ARCHITECTURE

In order to face the requirements outlined in previous section we have designed and built FameIR (see [1] for a detailed presentation of the system).

Basic capabilities of FameIR are the following:

- Indexing, retrieving and presenting multimedia documents.
- Querying in more than one collection at a time.
- Cross Language IR (CLIR) covering both, queries and documents, in Spanish, Catalan and English, using EuroWordNet (EWN[3]).
- Query expansion using EWN.
- Providing access to Web sites.

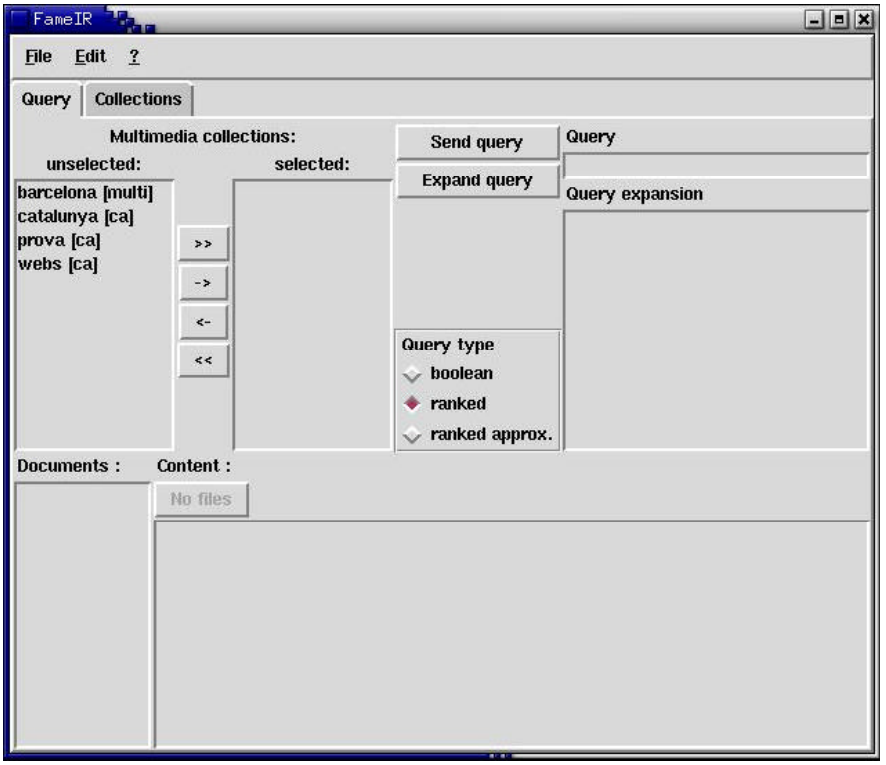

Figure 3. The initial window of the FameIR interface

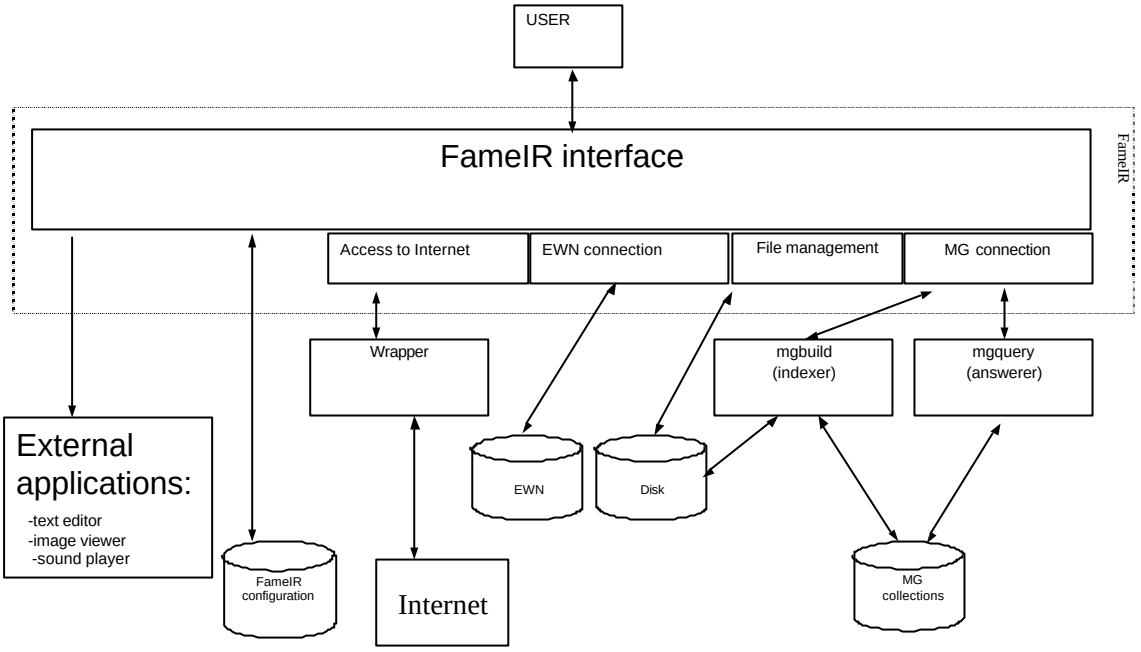

Figure 4. The architecture of FameIR.

The basic mode of performance of FameIR is by means of calls to functions stored in a Perl library. However, in order to achieve a better access to collections of multimedia documents, mainly during the indexing phase, a graphic interface (see Figure 3) has been built. The interface facilitates the management of different collections, the access to original documents of the collections and the operations of building and querying them.

FameIR has been built on the top of *Managing Gigabytes* (MG^2), (described in [11]), that provides the basic IR capabilities. Figure 4 presents the general architecture of FameIR. Users interact with the system through the FameIR interface. The core of the system is the MG connection that can be seen as a wrapper over MG software. All the capabilities of MG (basically indexing and querying) can be accessed in this way. A file management module provides facilities for managing MULTIMEDIA collections. EWN database, used for both query expansion and CLIR, can be accessed by the EWN connection module. Users can add new synonyms or translations of the terms of the query from those provided by EWN.

The system can access external applications for managing the different types of documents. The external applications are not part of FameIR but can be customised for accessing them. Among these external applications, a text editor, an image viewer and a sound player are currently available. For example, in case a new image document (**.gif** or **.jpg**) is added to the system, the image is displayed by the image viewer and a text editor window is offered to the user for input terms describing the image. In case of a voice document, the behaviour of the system depends on the attached files available. If the ASR (or transcription) file is available, its content is presented to the user as initial keyword file.

In FameIR, the set of Web sites that would be accessed during communication are previously selected and indexed. The indexing keywords associated with a Web source can be generated manually or automatically. The indexing keywords can be obtained automatically from the page title, the page URL, the keywords provided in the page (if there are any), the words associated with its address when it is referenced from other pages or a combination of this information. In the case of web services where pages are generated on the fly from some database, based on user request, they can be indexed either by words related to the service or by the each possible page. For example, the Web site *Classical Net*, providing information about classical

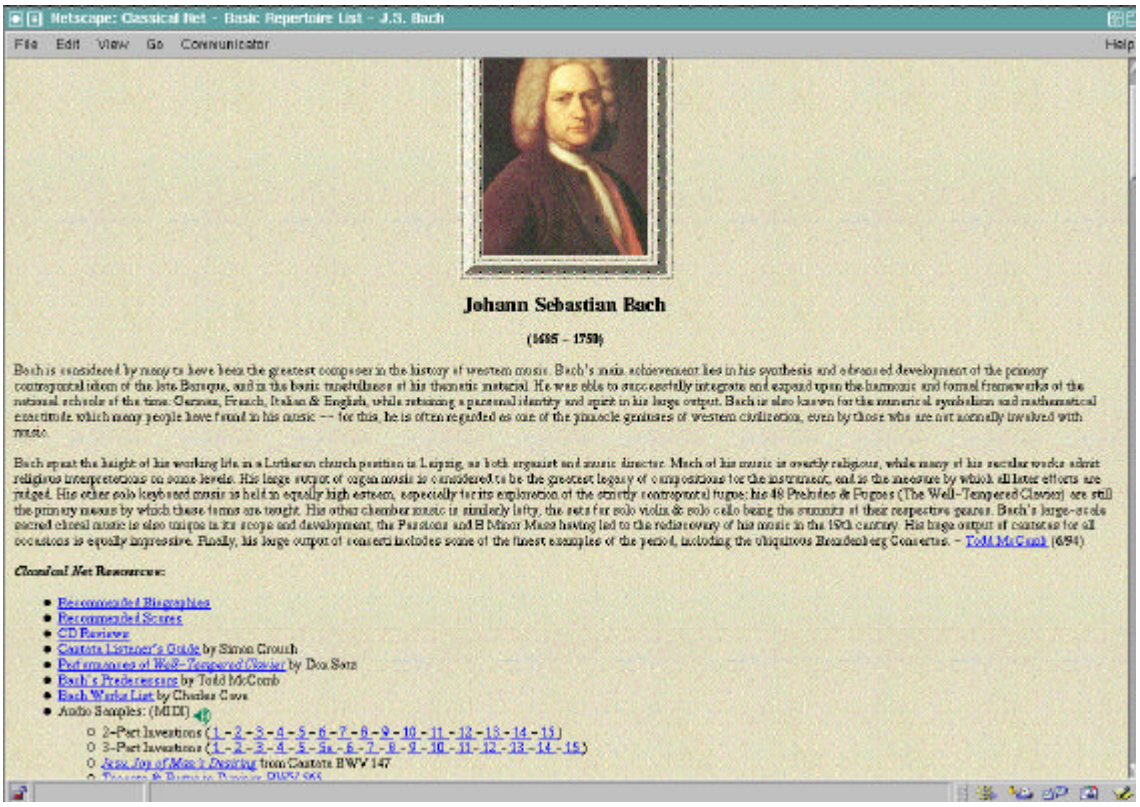

Figure 5. An example of Web page containing different types of data

composers (such as that about Bach shown in Figure 5), could be indexed by words general to the site (such as *composer*, *musician*, etc.) or by the name of all the composers described in it (i.e. *Bach*, *Brahms*, *Beethoven*, etc.).

The process of obtaining information from the Web sites is described in next section.

5. ACCESSING THE WEB SOURCES

The content of the Web sites is accessed during communication by wrappers. In the Web environment, a wrapper can be defined as a processor that converts information explicitly stored as in a HTML document into information explicitly stored as data structure for further processing.

We have adapted and enhanced the system of wrappers of GIWEB, a dialogue system for accessing web contents (described in [7]). This system of wrappers is designed to obtain information from different types of Web sources, such as corporation Intranets, public Intranets, Web services, personal pages, etc. In case of Web services where a set of parameters is required, this information is obtained from the terms in the user query (e.g. the departure and arrival cities in a train service or the person name in a biography service).

Wrappers can retrieve the whole web page (including its text, graphics, URL address, …) or a specific part of it, such as a table (or part of it) containing train schedules, a particular music file, all URL addresses, etc. In the FameIR system, the information retrieved by the system of wrappers is represented in HTML format or, alternatively, when retrieving multimedia files, in the format they appear in the Web source. For example, the Web page shown in Figure 5 containing different types of data (text, image, sound, URL address) could be retrieved as one Web document and also as several documents of different type: the text documents (a document can contain the whole text in the page or part of it, such as the two first paragraphs), the image file (in this page there is only one), the sound files (all of them or just a set of specific ones) and the web documents corresponding to the pages addressed (all of them or a subset of them).

There are several wrapper classes for dealing with different retrieval tasks. For example, there is a class in charge of obtaining all data in a specific column of a table, another class specialised in obtaining the image localised in a specific place in the page, etc.

In order to obtain the data, the wrappers represent the HTML page code as an HTML syntax tree where nodes are labelled by tag names. Then, following an explicit description of the type of data to be retrieved and its representation in the page (the path of tags to get it, the tags delimiting it, etc.) the wrapper traverses the HTML parsed tree and returns the obtained information. The file describing the data to be extracted can be created by the user manually, using the special language provided by the system. Currently, we are working in the automatic generation of the description file.

Because the information in the Internet is represented in heterogeneous sources that change rapidly, a lot of research has been dedicated to reduce the cost of obtaining the information necessary to extract data from a new site. Several approaches are being proposed to limit this cost. Basically, those approaches are based on providing semi-automatic tools (as those for visually generating wrappers described in [2]) and wrapper induction (as in the system described in [4]).

In our system, the knowledge necessary for obtaining the information from a new page, that is, the description of the representation of the data to be extracted, can be generated automatically using two different methods. The first method consists of using the description of the related pages. Usually, in a web site, pages describing similar contents follow similar organisation. The second method generates the description needed from one or more examples of the data to be retrieved. These examples can be provided easily by the user by selecting and copying them from the Web page (in a similar way of that described in [2]). Once the type of data and the information about its representation in the page (its situation in the HTML syntax tree representing the page, the tags delimiting the data and those contained in it,…) is automatically obtained, it is tested and the results are supervised by the user.

In the system of wrappers we have developed, if the data retrieved by the system when using an automatically generated description of the Web page is not exactly what the user expected, several strategies are applied to build new variations of this description. For example, if a web page contains a table and the user selects the data contained in the first column in the first row, the description file generated will allow the wrapper to obtain all the data in the first column of the table. In the case this result would not satisfy the user, a new description file will be generated which allows the wrapper to obtain the data in all columns of the table. If this is not the result desired, new strategies will be applied to generate other descriptions and thus allowing the retrieval of different information.

6. CONCLUSIONS

FameIR, a IR shell developed within the Fame project for providing extended IR capabilities has been presented.

FameIR allows management of multimedia documents: text (in several languages), graphics, voice, video, etc. It also provides query expansion and CLIR, using EWN. FameIR integrates as well a system of wrappers for accessing the Web and then indexing its content.

The system can be accessed both through a graphical interface and through a Perl library. We are currently working, within the Fame project, for allowing access to FameIR through a OAA[5] framework.

REFERENCES

[1] V. Arranz, M. Bertran, H. Rodriguez, "Computer Environment for Indexing and Retrieval of Multimedia Documents", Fame deliverable D7.1, September 2002.

[2] R.Baumgartner, S.Flesca and G. Gottlob. “Visual web information extraction with lixto”. Proceedings of VLDB, 2001.

[3] J. Choi, D. Hindle, J. Hirschberg, F. Pereira, A. Singhal and S. Whittaker (1999) “Spoken Content-Based Audio Navigation (SCAN)”. Proceedings of ICPhS-99, 1999.

[4] W. Cohen and L.S. Jensen, “A structured wrapper induction system for extracting information from semi-structured documents”, *Proceedings of IJCAI Workshop on Adaptive Text Extraction and Mining*, (2001).

[5] F. Crestani (2000) “Exploiting the Similarity of Non-matching Terms at Retrieval Time”. Journal of Information Retrieval, 2, 25-45, 2000.

[6] F. Crestani (2002) “Using semantic and

phonetic term similarity for spoken document retrieval and spoken query processing". In: B. Bouchon -Meunier, J. Gutierrez-Rios, R.R. Yager (eds.): *Technologies for Constructing Intelligent Systems*, pp. 363-376, Springer-Verlag, Heidelberg, Germany, 2002.

[7] M. Gatius, H. Rodríguez. "Natural Language Guided Dialogues for Accessing the Web. In the Proceedings of the *Fifth International Conference on Text, Speech and Dialogue.* Brno, Czech Republic,2001. Springer-Verlag in Lecture Notes in Artificial Intelligence subseries of LNCS series as Volume 2448.

[8] H. Joho, C. Coverson, M. Sanderson and M. Beaulieu (2002) "Hierarchical presentation of expansion terms". Proceedings of ACM SAC, 2002.

[9] M.T. Maybury, ed. (1997) *Intelligent Multimedia Information Retrieval.* MIT Press, 1997.

[10] P. Schäuble (1997) *Multimedia Information Retrieval,* Kluwer Academic Publishers, 1997.

[11] A. Singhal and F. Pereira (1999) "Document Expansion for Speech Retrieval". ACM SIGIR'99, pp. 26-33, 1999.

[12] S. Whittaker, J. Hirschberg, J. Choi, D. Hindle, F. Pereira and Amit Singhal (1999) "SCAN: Designing and Evaluating User Interfaces to Support Retrieval from Speech Archives". ACM SIGIR'99, 34-41, 1999.

[12] I.H. Witten, A., Moffat, Alistair and T.C. Bell (1999) *Managing Gigabytes: Compressing and Indexing Documents and Images.* Morgan Kaufmann Publishing, San Francisco, 1999.

[14] J.K. Wu, M.S. Kankanhalli, J-H Lim, D. Hong (2000) *Perspectives on Content-Based Multimedia Systems*, Kluwer Academic Publishers, 2000

---

[1] The FAME (Facilitating Agent for Multicultural Exchange) project develops a new vision for computer interfaces, which replaces and extends strictly human-computer interaction by computer-enhanced human-to-human interaction: http://isl.ira.uka.de/fame/

[2] Managing Gigabytes: http://www.cs.mu.oz.au/mg/

[3] A generic wide coverage multilingual database with WordNets for several European languages. English, Catalan and Spanish are used in FameIR: http://www.let.uva.nl/~ewn/

[4] Classical Net. http://www.classical.net

[5] Open Agent Architecture: http://www.ai.sri.com/~oaa/