A System for Supporting Cross-Lingual Information Retrieval[♣]

Joanne Capstick, Abdel Kader Diagne, Gregor Erbach, Hans Uszkoreit
German Research Center for Artificial Intelligence
Language Technology Lab
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany

Anne Leisenberg, Manfred Leisenberg
Bertelsmann Online-Media-Service
Carl-Bertelsmann-Str. 161 0
33311 Gütersloh
Germany

http://mulinex.dfki.de/
mulinex@dfki.de

Abstract

In this paper, we present the system MULINEX, a fully implemented system which supports cross-lingual search of the WWW. Users can formulate, expand and disambiguate queries, filter the search results and read the retrieved documents by using only their native language. This multilingual functionality is achieved by the use of dictionary-based query translation, multilingual document categorisation and automatic translation of summaries and documents.
The system supports French, German and English and has been installed and tested in the online services of two European internet content and service provider companies.
This paper focuses on the techniques and algorithms used in the MULINEX system, explaining how each component works and how it contributes to the overall functionality of the integrated system. The primary system functionalities are outlined from the user perspective, followed by a description of the document database used in the system. The technologies and linguistic resources used in the various system components are then described in detail.

[♣] **Acknowledgements:** The work described in this paper was financially supported by the European Union's Telematics Application Programme, contract LE-4203 in the Language Engineering sector. Thanks to the EU project officers, the entire MULINEX team, and two anonymous reviewers for their support, enthusiasm and constructive criticism.

1 Introduction

With the steady increase of internet users outside the US, English is losing its dominant position in the internet and we are witnessing the emergence of a truly multilingual medium. The interest in supporting a multilingual internet is demonstrated through web internationalisation initiatives and through the addition of language technologies, such as language identification and machine translation by leading search engines. Cross-lingual retrieval takes this trend further and provides the means for accessing multilingual internet content, by enabling queries made in one language to retrieve documents in one or more other languages.

In this paper, we present the system MULINEX, whose objective is to enable users to search in multilingual document collections using their native language, supported by an effective combination of linguistic and information retrieval technologies.

Both monolingual and cross-lingual full-text retrieval are faced with the problem of understanding the actual intention of a user query. Two complementary strategies can help alleviate this situation: firstly, query formulation support to aid the user in making more focussed queries, and, secondly, tools for filtering and navigating through search results, providing users with accurate and efficient access to those documents which satisfy their information needs. For dictionary-based query translation, support for interactive query translation disambiguation is crucial to avoid a loss of precision through inaccurate translations.

The MULINEX system described in this paper combines current information retrieval technology with state-of-the-art language technologies. The system emphasises user-friendly interaction, which supports the user by offering query translation and expansion, by presenting search results along with information about language, thematic category, automatically generated summaries, by allowing the user to filter results according to multiple criteria. The basic components are embedded in an object-oriented, manager-based architecture, providing a flexible system with potential for extendibility and re-usability.

This paper focuses on the techniques and algorithms used in the MULINEX system,[1] explaining how each component works and how it contributes to the functionality of the overall integrated system. The primary system functionalities are outlined from the user perspective, followed by a description of the document database. The technologies and linguistic resources used in the various system components are then described in detail.

2 Functionality for the User

MULINEX is aimed at users who want to retrieve information from the WWW which may be represented in web pages in different languages. Users need not have any knowledge of the foreign language, since the cross-language retrieval process is fully supported by translation of queries, of summaries and of retrieved documents. However, the system is equally useful for users with some knowledge of the foreign languages, since it provides convenient support for query translation, and allows filtering of the search results by language and thematic categories.

A user requirements study was carried out at the beginning of the project. The study consisted of questionnaires which were filled out by users of internet service and content providers in Germany and France (Hernandez, 1997). The survey revealed that most users

[1] (Erbach et al., 1998) provides more detailed information on the social and economic factors influencing the project, the objectives of the project consortium members, and the user requirements for the system.

consider restriction of the search search according to thematic category and language to be important features, and that translation of search results, informative presentation and personalisation of the system were considered as useful

A psychological experiment with 84 subjects was carrried out based on a mock-up version of the system (Capstick et al., 1998). We used a 3 x 2 setup in which six groups of users were presented with different organisation of results (by thematic category and by relevance with respect to the query terms) in combination with different summary types (first 200 characters, extracted HTML headings and emphasised text, query specific summary). All subjects had the option to access machine translations of the summaries. The subjects were given fixed queries and were asked to identify relevant documents from the search results. In a survey based on a questionnaire, the users expressed positive reactions about the ordering of search results by thematic category. However, there were no statistically significant differences in user performance over the six experimental conditions.

Based on the results of the user requirements analysis, the system provides the following functionality to support the user in retrieving documents from multilingual document collections:

- translation of the user's query
- interactive disambiguation of the query translation (optional)
- interactive query expansion (optional)
- simultaneous search in English, German and French document collections
- informative presentation of search results, with summary, language and thematic category
- filtering of search results by language and category
- on-demand translation of summaries and search results

Cross-language retrieval research started with Salton's seminal paper (Salton, 1973), and has become an active field of research over the past four years (Hull and Oard, 1997; Yang et al., 1997; Greffenstette, 1998). In terms of Oard's classification of cross-language retrieval approaches (Oard, 1997a), the query translation approach adopted in our system is a knowledge-based approach, as opposed to the corpus-based approach based on comparable corpora adopted in Sheridan and Ballerini (1996) and to approaches which construct parallel corpora by means of automatic document translation (Oard, 1997b; Kraaij and Hiemstra, 1997; Hiemstra and Kraaij, 1998).

The query translation approach is used because of the lack of substantial parallel or comparable multilingual corpora[2], and because the document translation approach is not scalable to very large amounts of data because of the resource requirements of currently available machine translation systems.

We will now illustrate how the user interacts with the system. Queries are formulated by keywords as in a standard WWW search engine (see the search box in the top of figure 1). Since automatic language identification of short queries is error-prone, the query language must be specified by the user. The user can also select the acceptable document languages.

The user interface is available in English, German, and French, and is extensible to other languages. Users can switch the user interface language at any time during the interaction.

In the next step, the query is translated into the selected target languages. Since search engine queries typically do not provide enough context for automatic disambiguation[3], the

[2] Parallel corpora contain translated documents, while comparable corpora contain texts which are not translations, but talk about the same topic (e.g., two newspaper articles about the same event written by journalists in different countries).

[3] Our examination of 100 000 queries submitted to the German web search engine web.de in 1998 revealed an average query length of 1.3 words.

"query assistant" provides the opportunity for interactive disambiguation of the query translations (see figure 1 for the translations and expansions of the query term *fair*). In order to help users who do not understand the target language with the disambiguation of query translations, the "query assistant" shows how each translated query term translates back into the original query language. As the example for the query term *fair* in figure 2 shows, the back translations assist the user in eliminating translations into German and French which are irrelevant to the intended meaning[4] even though the user may not have any knowledge of German and French. The precision of the German and French queries is thus improved.

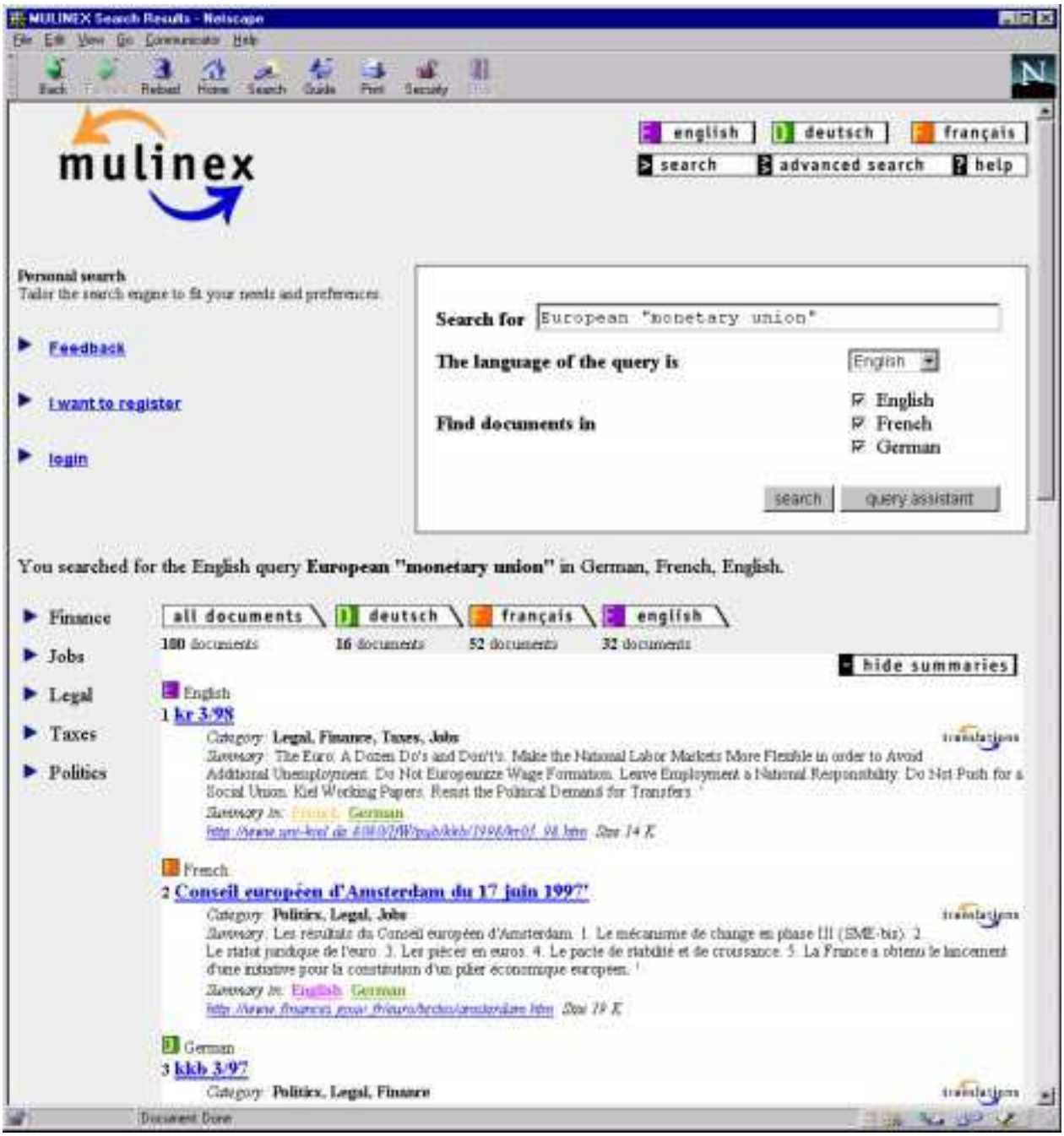

Figure 1: Query Form and Search Results Presentation

[4] Of course this method will only work to the extent that the intended meaning of the translation of the query term has alternative paraphrases in the original query language. The effectiveness of the method is improved if the underlying dictionary lists the most common translations before rare or domain specific translations.

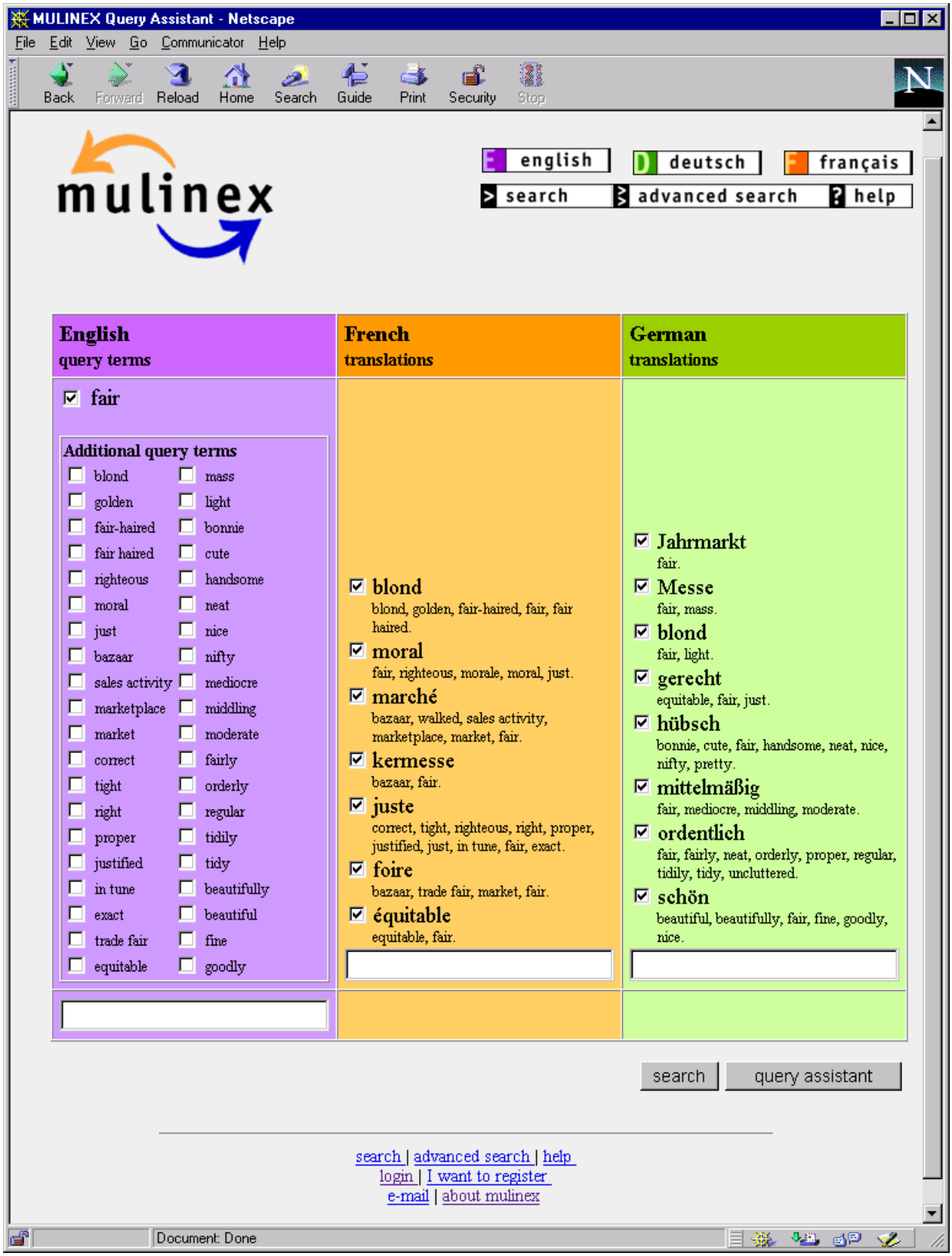

Figure 2: Query Assistant

The paraphrasing of the query term which results from the query translation and back-translation is also used to provide a simple query expansion mechanism which suggests alternative query terms to the user in the original query language. In the example shown in figure 2, the query terms *trade fair* and *sales activity* could also be selected to expand the

query in the original language (English). After query expansion, disambiguation may optionally be repeated. Following query translation and expansion, the documents for each language are searched in parallel with the search terms for this language.

Figure 1 shows how search results are presented to the user. The search form with the original query and options is presented on the results page. The results list contains documents in all languages requested by the user and is sorted by relevance. For each document in the list, its language, title, URL and size are displayed. The document categories are presented in the user interface language and the summary in the document language. Users may request a translation of the summary, which is displayed in a separate window. The translation icon on the right provides automatic document translation.

By selecting the corresponding language tab, the results list can be filtered by language. The category navigation tool on the left hand side of the page enables the user to filter the results by category.

In the following sections, we will specify what information is stored about each document in order to support the filtering and presentation of the search results, and the technologies and linguistic resources used to achieve the functionality described, and to obtain the information about each document.

3 Document Database

The core of the system is a database in which certain pieces of information about all documents are stored. In the context of our search engine, a *document* is a unit of presentation that is accessed by a WWW user by following a hyperlink. A document may be composed of several web pages which are arranged in a frameset. Treating the entire frameset as one document has the advantage that queries made up of several terms can retrieve a frameset in which the terms occur in different frames. For example, the query *travel thailand* may retrieve a document in which the query terms occur in separate frames. Another advantage is that retrieved frames are presented in the context of their frameset.

Documents in which multiple languages occur are not handled explicitly. Although the language identification module could well identify different languages in a document if it were run separately for each paragraph (or other suitable unit) of a document, we assign only one language on the basis of the text which occurs at the beginning of the document. This decision was taken in order to speed up the document analysis process and because a retrieval system for the WWW cannot easily retrieve (or refer to) a passage or paragraph of a document which is written in a specific language.

In order to provide the functionality outlined in section 2, the following information is stored about each document:

URL	Uniform Resource Locator of the Document
Title	title of the document, as specified in the HTML <TITLE> tag
Size	size of the document, as provided in the HTTP protocol
Date	last modification date of the document, as provided in the HTTP protocol
Language	language of the document (see section 4.2)
Keywords	keywords, author-specified or automatically extracted (see section 4.3)
Summary	summary, author-specified or automatically extracted (see section 4.3)
Categories	a list of categories and similarity values (see section 4.4)
Full Text Index	full text of the document, indexed for document retrieval

Table 1: Structure of a record in the document database

The Fulcrum SearchServer, a state-of-the-art document management and retrieval system with an SQL-based query language, is used as the document database. Depending on the intended use of the system and availability of disk space, the full text of the documents can be stored in addition to the full text index.

4 Technologies and Resources for Document Analysis

In this section we describe the technologies and linguistic resources which are used to analyse documents in order to obtain the information specified in table 1. Document analysis takes place during the gathering of documents by means of a web spider.

4.1 Document Gathering

Like all WWW search engines, MULINEX makes use of a web spider for the acquisition of documents and of a core information retrieval system for supporting the search. MULINEX extends this basic functionality by performing additional document analysis steps.

Figure 3 shows the steps of the document acquisition process.[5]. At each step, the information about a document is successively refined. The web spider obtains information that is specified in HTTP and HTML such as size, modification time, the URL, the character encoding, and the full text of the document. The document analysis components analyse the content of the document to determine the language and thematic categories, and to create a document summary. All this information is then used to create or update a record in the document database.

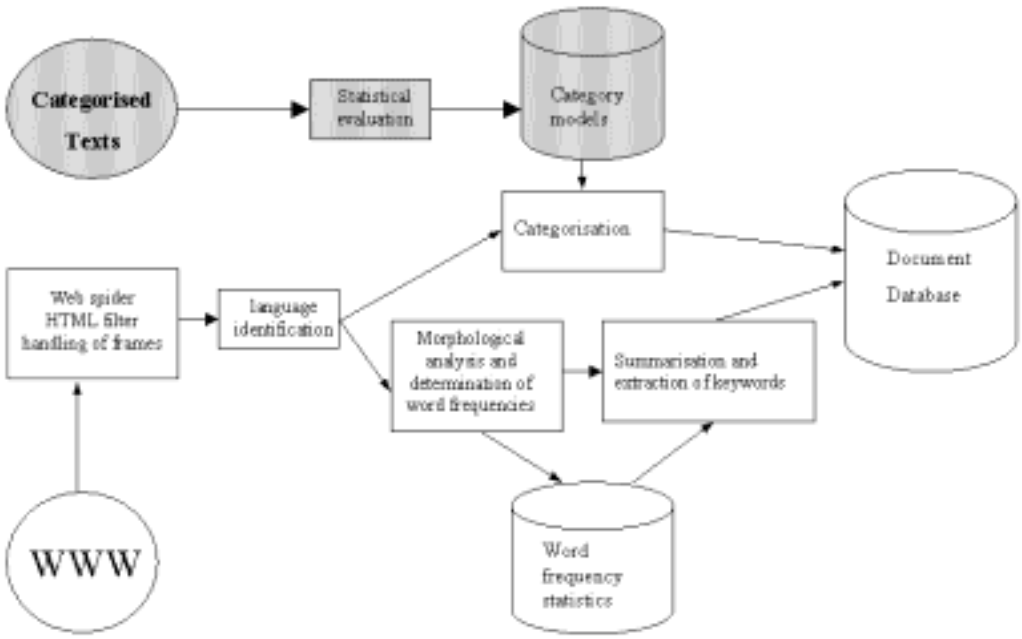

Figure 3: Document Acquisition

[5] Preprocessing steps which take place prior to document acquisition are shown with a shaded background

Gathering of documents is performed by the Harvest gatherer (Bowman et al., 1994), a highly configurable system for gathering documents from the WWW which respects the Robot Exclusion Protocol.

Harvest consists of two parts: the gatherer (a web spider) and the broker (an indexing and search system based on the Glimpse retrieval engine). In the MULINEX system, we have decided to use only the gatherer and replace the broker by the Fulcrum SearchServer. Fulcrum provides certain advantages over Glimpse as a retrieval engine, notably an SQL-based query language and the capability to sort search results by relevance.

4.2 Language Identification

Information about the language of a document is a necessary prerequisite for further processing steps: document categorisation, summarisation, and machine translation are all dependent on knowing the document language. Knowledge of the language also improves indexing and retrieval performance by using appropriate stop-lists, stemming, term weights, thesauri, etc. for each language. The language of WWW documents is often not marked by the author even though HTML and HTTP allow the author to provide this information. Therefore, we use a statistical language identifier to determine the language of a document.

Language identification is performed by making use of an algorithm which compares the relative frequencies of the most frequent n-grams (from 1 to 5 characters) in a document to 40 stored language models (Cavnar and Trenkle, 1994).

For each language, the language recogniser uses a language model - the sequence of the 300 most frequent n-grams, ordered by their frequency in a training corpus.[6] For each document whose language we want to identify, we compute the sequence of its most frequent n-grams. For each n-gram of the document, we compare its rank to the rank of the same n-gram in the language model, and sum up the differences. A maximal difference value is used for n-grams which are not present in the language model. The language whose language model has the smallest total difference to the current document will be assigned.

The language identifier was evaluated using data from the European Corpus Initiative CD-ROM in Danish, Dutch, English, French, German, Italian, Norwegian, Portuguese, and Spanish, replicating the experiments reported in Greffenstette (1995), which compared language identification schemes based on trigrams and on frequent short words. The evaluation results of our n-gram language identifier and a comparison to Greffenstette's results for the MULINEX languages English, German and French are given in table 2.

It has been argued that a language identifier based on the frequencies of n-grams from length 1 to 5 combines the advantages of several well-known language identification methods: it takes into account the frequencies of single letters, of bigrams and trigrams and of frequent short words. However, our experiments showed significant advantages for French text of a length between 6 and 15 words only. Above 21 words, all algorithms show an almost perfect language identification accuracy.

A separate test showed that the language identification performance for the ECI corpus data did not degrade when the distinction between upper and lower case characters was ignored. For language identification of web pages, we ignore case because capitalised titles and headings often lead to errors in language identification.

[6] We used the European Corpus Initiative (ECI) CD-ROM to obtain our training data.

	Method	3 to 5 words	6 to 10	11 to 15	16 to 20	21 or more
English	N-Gram	68.75	97.1	99.3	99.7	100
	Trigram	97.2	99.5	99.9	99.9	100
	short words	87.7	97.3	99.8	99.9	100
German	N-Gram	95.1	98.4	99.2	99.9	100
	Trigram	97.2	99.3	99.8	99.9	100
	short words	71.6	89.6	98.2	99.8	100
French	N-Gram	84.9	98	98	98.3	99.5
	Trigram	93	94.5	93.6	99.8	100
	short words	81.8	96	97.2	99.8	100

Table 2: Evaluation of three language identification methods (see text)

4.3 Summarisation

Summarisation is performed by selecting the sentences[7] which best characterise a document. We use sentence selection as our summarisation method because it shows robust performance on a collection of widely varying documents, such as the WWW.

There are two summarisers, a neutral (query-independent) and a tailored (query-dependent) summariser. Both work by selection of the most salient sentences. The target length of the summary can be specified, and the summariser will select the most important sentences until the target length has been reached. The query-independent summariser selects sentences which are marked up by structural (headings) and layout-oriented (boldface, italics) markup, and uses heuristics such as selecting the first sentence of a paragraph and making use of term frequency statistics. The query-specific summariser selects sentences in which the query terms (or morphological variants) occur.

In the MULINEX System, we use the query-independent summariser during document gathering to generate summaries which are stored in the document database. Use of a query-specific summariser for the search system is not practical, as we do not store the full text of the documents on the server.

In addition to sentence extraction, we extract a set of salient keywords for each document by choosing words which occur frequently in the document, but less frequently in the document collection.

4.4 Categorisation

The MULINEX system contains three different document categorisation algorithms, each suited for different categorisation tasks:

1. n-gram categoriser for noisy input
2. k-nearest-neighbour (KNN) algorithm for normal documents
3. pattern-based categoriser for very short documents

The n-gram categoriser makes use of the frequencies of n-grams of characters in a document which is compared to a category model, the sequence of most frequent n-grams ordered by

[7] We treat words between punctuation or structural HTML markup as sentences, even if they are not grammatically well-formed sentences.

their frequency in a training corpus.[8] Our evaluation has shown that this categoriser did not perform as well for web pages as the KNN categoriser. The n-gram categoriser is more useful in situations with noisy input, such as categorising OCR output.

The k-nearest-neighbour algorithm (Yang, 1994) is a statistical algorithm which classifies a new document by combining the category assignments of the k most similar training documents, weighted by the similarity between the new document and each of the k best matching training documents. The categoriser has been trained with documents from newsgroups in French, German and English.

Although the categorisers are trained separately for each language, multilinguality is achieved by training for the same categories in different languages. For example, we gathered training material for the category "politics" from the German newsgroup *de.soc.politik*, the correspondingFrench group *fr.soc.politique* and English *soc.politics*. For medicine, we used all English *sci.med.** groups, German *de.sci.medizin* and French *fr.bio.medecine*.

The two statistical categorisers (KNN and N-gram) have been evaluated with German news articles which were obtained from the Federal Press Office of the German government. The task of the Press Office is to distribute news articles to the different departments and ministries of the German government. The training data were categorised according to the government department to which they were distributed. We had a total of 8809 documents with 73 distinct categories; a document can have more than one category if it is sent to more than one government department. The categorisers assigned a number of categories with a confidence value; all category assignments above a fixed cutoff point, which was chosen to maximise the f-measure value on the training set for each category, were used in the evaluation. 80 % of the documents were used for training, the rest was retained for testing. Precision and recall were evaluated for each of the 73 categories with the following results.

		KNN	**N-gram**
Recall	Average	0.757	0.243
	Standard Deviation	0.217	0.237
Precision	Average	0.714	0.361
	Standard Deviation	0.173	0.224

Table 3: Precision and recall for two categorisation algorithms

In addition, there is a pattern-based categorisation algorithm for narrow, specialised categories. The pattern-based categoriser recognises terms which are of interest to a certain domain, such as the names of travel agencies or airlines for the tourism domain. It is used in situations where the pages to be categorised contain only little text and are dominated by graphics and tables. The patterns, expressed in Perl regular expression syntax, have been defined manually, by abstracting over multi-word terms found in a corpus of web pages for the domain of interest.

The pattern-based categoriser has not been formally evaluated due to the lack of categorised training data, but usability testing of the system showed user satisfaction with categorisation results (cf. section 6).

[8] This is the same algorithm that is used for language identification. It has been used successfully by Cavnar and Trenkle for the categorisation of Usenet news articles (Cavnar and Trenkle, 1994).

5 Technologies and Resources for Search and Result Presentation

In this section, we describe the technologies and resources used to process a user's query and present the search results.

5.1 Query Analysis

The MULINEX system analyses, translates and expands the users' queries. Since the retrieval performance of automatically translated queries is inferior to monolingual information retrieval, there is an (optional) step of user interaction, where the user can select terms from the translated query and add other translations.

The search syntax supports required search terms (+), excluded search terms (-), and allows the user to block the translation of search terms (!). Full boolean search syntax (AND/OR) is not supported because of the problems encountered with ambiguity in the translation of query terms.

Queries are morphologically analysed by making use of MMORPH (Petitpierre and Russell, 1995), and then translated by making use of multilingual dictionaries. The translated queries are the input to the search in the document collection. The search is performed separately for each language in order to avoid retrieving irrelevant documents because of accidental cross-language homonymy (e.g., when the Italian query *cute* (skin) retrieves English pages containing the word *cute*). A separate index exists for each language, so that the appropriate stopwords and terms weights are used.

5.2 Query Translation

Queries are translated by lookup of the analysed query terms in bilingual dictionaries. Multilingual lexical resources are used for query translation and query expansion. The MULINEX system uses six bilingual dictionary databases with 100.000 to 200.000 entries each for all six language pairs supported by the system (German-English, German-French, French-English and the converse pairs). An entry is a pair of a source language term and a target language term. So, if a term has five translations, we count five dictionary entries.

The multilingual lexicons are based on resources obtained from the European Union (the Euterpe database from the European parliament, and the Eurodicautom from the EU's translation service), which have been combined and augmented with lexical resources from the public domain. The dictionaries contain both single-word terms and multi-word phrases and collocations. The translated and expanded queries are submitted as SQL queries to document database.

5.3 Result Presentation and Machine Translation

The presentation of search results shown in figure 1 is achieved by means of a presentation server which constructs the presentation from a multilingual presentation database. Additional presentation languages can be added by making additions to this database. Category names are always shown in the presentation language, regardless of the language of the document.

Machine translation is used for the automatic translation of summaries and web pages. The LOGOS machine translation system is used to provide translation services in MULINEX.

6 User Evaluation

The evaluations of individual system components have been described in the previous sections. In this section we summarise the results of usability testing which has been carried out in the first quarter of 1999. During this time, the system was installed in the online service of Bertelsmann Telemedia with a database of ten thousand travel and tourism documents, and was accessible for all internet users. The documents were automatically categorised with the categories "last minute", "accommodation", "transport", "adventure travel" and "tour operators", using the pattern-based categoriser with patterns which were produced manually by generalising over examples obtained from a categorised corpus. The evaluation was carried out through questionnaires which were filled in by 52 users of the system, selected from customers of the commercial online service CityWeb. 98 per cent of the respondents were native speakers of German, 25 per cent female, 67 per cent under 30 years old and 46 per cent had a university degree.

56 per cent of the respondents were fully satisfied with the categorisation, and 40 per cent were partially satisfied. 52 per cent were fully satisfied with the query translation, and 46 per cent partially satisfied. 96 per cent judged the user interface as intuitive, and only 4 per cent as complex. 75 per cent were fully satisfied with the speed of the search, but only 40 per cent were fully satisfied with the search results. This low figure can be explained by the fact that the document database contained only a very limited number of entries. The user judgements of the usefulness of the system's core features are shown in table 4.

	Categorisation	**Query translation**	**Summaries**	**Summary translation**
not satisfied	2	1	5	10
somewhat satisfied	21	24	17	15
fully satisfied	29	27	30	27

Table 4: User judgements of core system features

In the field for free comments, a few of the users expressed their dissatisfaction with the quality of automatic translations of summaries and documents.

7 Implementation

The system has been implemented in Java and Perl, based on an object-oriented and manager-based architecture (Diagne, 1998). The benefits of an object-oriented architecture are increased modularity, a flexible system and ease of reusability through inheritance of structures and components (Jacobson et al., 1992). In addition to the functionality described above, the system allows users to register with the system and save their preferences. Registered users can make use of an agent system to perform searches periodically and be notified when new documents become available. For a detailed technical description of the system, see Capstick et al. (1999).

8 Conclusion and Future Work

As far as we are aware, MULINEX was the first system to support multilingual search by using only one's native language through translation of queries and summaries. Our ongoing work is concerned with the addition of other languages, a personal agent system for registered users, clustering of search results and visualisation of clusters.

In the current system, no domain modelling is used. The performance of some system components (query translation and document translation) can be performed if the system is adapted to a specific domain.

We are working on extending the system into an information management platform by extending the document database to store person, company and place names, which are obtained by means of information extraction technologies.

References

Bowman, C.M., Danzig, P.B., Hardy, D.R., Manber, U., Schwartz, M.F. and Wessels, D.P. (1994) Harvest: A Scalable, Customizable Discovery and Access System. Technical Report CU-CS-732-94, Department of Computer Science, University of Colorado, Boulder, August 1994

Capstick, J., Erbach, G. and Uszkoreit, H. (1998) Design and Evaluation of a Psychological Experiment on the Effectiveness of Document Summarisation for the Retrieval of Multilingual WWW Documents. Working Notes of the AAAI Spring Symposium "Intelligent Text Summarisation". Stanford.

Capstick, J., Diagne, A.K., Erbach, G. and Uszkoreit, H. (1999) MULINEX: An Integrated System for Cross-lingual Retrieval and Interactive Refinement of Multilingual Queries. Technical Report, German Research Center for Artificial Intelligence, Saarbrücken.

Cavnar, W.B. and Trenkle, J. M. (1994) N-Gram-Based Text Categorization. Paper presented at the Symposium on Document Analysis and Information Retrieval. Las Vegas.

Diagne, A.K. (1998) Die MORE Architektur - ein objekt-orientiertes Architekturmodell für komplexe und netzwerkfähige Softwaresysteme. Paper presented at the workshop "Deklarative KI-Methoden zur Implementierung und Nutzung von Systemen in Netzen" at KI-98. Bremen.

Erbach, G., Capstick, J., Diagne, A.K., Uszkoreit, H., Cagno, F., Gadaleta, G., Hernandez, J.A., Korte, R., Leisenberg, A., Leisenberg, M. and Christ, O. (1998) MULINEX: Multilingual Web Search and Navigation. Paper presented at the Conference Industrial Applications of Natural Language Processing, Moncton.

Grefenstette, G. (ed). (1998) Cross-Language Information Retrieval. Kluwer, Boston.

Grefenstette, G. (1995) Comparing Two Language Identification Schemes. In the proceedings of 3rd International Conference on Statistical Analysis of Textual Data (JADT'95), Rome.

Hernandez, J. A. (1997) MULINEX User Requirements: Synthesis Report. MULINEX deliverable report 2.3, Grolier Interactive Europe, Paris.

Hiemstra, D. and Kraaij, W. (1998) Twenty-One in ad-hoc and CLIR", In: Proceedings of the Seventh Text Retrieval Conference (TREC-7), E.M. Voorhees, D. K. and Harman, D. K. (eds.), NIST special publication 500-240.

Hull, D. and Oard, D. (eds.) (1997) Cross-Language Text and Speech Retrieval - Papers from the 1997 AAAI Spring Symposium, AAAI Press, Menlo Park.

I. Jacobson, M. Christerson, P. Jonsson and G. Övergaard. (1992) Object-Oriented Software Engineering – A Use Case Driven Approach. Addison-Wesley, Reading, MA; ACM Press, New York, 1992.

Kraaij, W. and Hiemstra. D. (1998) Cross Language Retrieval with the Twenty-One system, In: Proceedings of the Sixth Text Retrieval Conference (TREC-6), Voorhees, E.M. and Harman, D. K. (eds), NIST Special Publication 500-240, pp. 753-761.

Oard, D. (1997a). Alternative Approaches for Cross-Language Text Retrieval. Paper presented at the AAAI Spring Symposium on Cross Language Text and Speech Retrieval. Stanford.

Oard, D. (1997b) Document Translation for Cross-Language Text Retrieval at the University of Maryland. Paper presented at the Sixth Text Retrieval Conference (TREC-6), Gaithersburg.

Petitpierre, D. and Russell, G. (1995). MMORPH – The Multext Morphology Program. Multext deliverable report for the task 2.3.1, ISSCO, University of Geneva, February 1995.

Salton, G. (1973) Experiments in Multi-Lingual Information Retrieval. Information Processing Letters, 2(1), 6-11.

Sheridan, P. and Ballerini, J.P. (1996) Experiments in Multilingual Information Retrieval using the SPIDER System. Paper presented at the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Zürich.

Yang, Y. (1994). Expert Network: Effective and efficient learning from human decisions in text categorization and retrieval. Paper presented at the 17th ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 13 – 22. Dublin.

Yang, Y., Carbonell, J. G., Brown, R. and Frederking, R. E. (1998) Translingual Information Retrieval: Learning from Bilingual Corpora. Artificial Intelligence Journal, special issue: Best of IJCAI-97, pp323-345.