First publ. as paper in: Informationskompetenz - Basiskompetenz in der Informationsgesellschaft, Proceedings des 7. Internationalen Symposiums für Informationswissenschaft (ISI 2000), Darmstadt, 8.-10. November 2000, pp. 127-143

INSYDER - Information Retrieval Aspects of a Business Intelligence System

Gabriela Mußler, Harald Reiterer, Thomas M. Mann

Universität Konstanz
Fachbereich Informatik und Informationswissenschaft,
D-78457 Konstanz
{Gabriela.Mussler, Harald.Reiterer, Thomas.Mann}@uni-konstanz.de

Zusammenfassung

Dieser Beitrag beschäftigt sich mit INSYDER, einem visuell unterstützten Suchsystem im Umfeld der sogenannten Business Intelligence Systeme. Dabei liegt der Schwerpunkt dieses Beitrags in der Erläuterung der Information Retrieval- und Visualisierungsaspekte zur Unterstützung des Suchprozesses im WWW. Vorgestellt wird das Rankingverfahren, das Interaktive Relevance Feedback sowie die Beschreibung der Unterstützung des Benutzers bei der Formulierung der Suchanfrage mittels Visualisierung.

Abstract

This paper presents a visual information seeking system for the WWW called INSYDER[1]. The aim of INSYDER is to find business information from the WWW. Information seeking - especially on the Web - is an imprecise process. Information seekers often have only a fuzzy understanding of how they can get the information they want. This paper focuses on the use of information retrieval and visualisation techniques of the INSYDER system to support the user in extracting information from the WWW, namely in formulating queries, refining and reviewing results.

[1] The research project INSYDER (INternet SYstème DE Recherche) was funded by a grant from the European Union, ESPRIT project number 29232.

1 Introduction

The benefits of using external information for business intelligence systems[2] are significant. An enterprise must know more and more about its customers, its suppliers, its competitors, government agencies, and many other external factors. Valuable information about external business factors is readily available on the WWW and its amount is increasing every hour. While a few WWW resources are used as data sources, the immense resources of the Internet are largely untapped. What is needed is a continuous and systematic approach to make use of these untapped resources. [Hackathorn 1998] proposes such an approach called Web farming: *"Web farming is the systematic refining of information resources on the Web for business intelligence."* The visual information seeking system INSYDER presented in this paper can be seen as a kind of Information Assistant [Kuhlen 1999] for finding and analysing business information from the Internet.

A detailed description of the INSYDER system can be found in [Mann, Reiterer 1999], [Mußler 1999] and first results of a user evaluation of the different visualisations are presented in [Reiterer, Mußler, Mann et al. 2000], [Mann, Reiterer 2000]. This paper focuses on the use of information retrieval and visualisation techniques to support the user in extracting information from the WWW, namely in formulating queries, refining and reviewing results. Here we shall present ideas and their prototypical implementation extending and continuing the original idea of INSYDER. These ideas were mainly born when reviewing results of various surveys concerning information seeking on the WWW. Conclusions of these surveys are that users often do not know how to express their information need [Nielsen 1997], [Pollock, Hockley 1997], that Relevance Feedback is only useful, if made transparent to the user [Koenemann, Belkin 1999], and that users have problems with the current paradigm of information retrieval systems simply presenting long lists of results [Zamir, Etzioni 1998]. In the following, we present our proposed solutions to these

[2] "A business intelligence system ... provides a set of technologies and products for supplying users with the information they need to answer business questions, and make tactical and strategic business decisions." [IBM]

problems, which have been included in the overall Information Assistant approach of INSYDER[3].

The paper is organised into the following chapters: Chapter two gives a short overview of the architecture and main components of the INSYDER system. Chapter three describes the novelty of the INSYDER system. Chapter four describes the enhancements to the existing system, which are partly finished, namely a query interface, called Visual Query, the two used ranking algorithms used in INSYDER, and the pursuit of the Relevance Feedback. Chapter five gives a summary of the ideas of this paper and an outlook on future work.

2 Overview of the INSYDER system

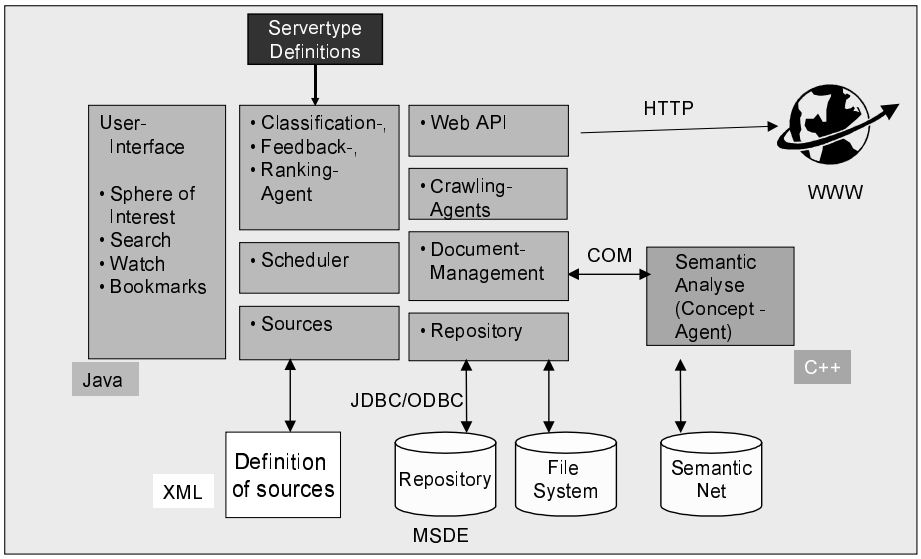

Figure 1: Architecture of the INSYDER system

The INSYDER Information Assistant acts on the user's behalf and is built up using different agents using Information Retrieval techniques and a synchronised visualisation approach.

Figure 1 shows the architecture of the INSYDER system consisting of several components mainly developed in Java. Only the component for the semantic

[3] After the official EU-project has been finished in March, the University of Konstanz is still working on the INSYDER system in close relation with the French company Arisem.

analysis has been developed in C++, because it is a further development of an existing module by the project partner Arisem already written in C++.

The **user-interface** and all visualisations (e.g. Scatterplot, BarGraph, TileBar, Relevance Curve) have been developed in Java using the JFC (Java Foundation Classes) also called Swing.

The different **agents** are responsible for special retrieval tasks (e.g. crawling the WWW; clustering and ranking of the search results; preparing the relevance feedback for a new crawling).

The **scheduler**'s is responsible for the monitoring process of the WWW (watch function). The watch function is able to check user-defined Web pages for changes regularly.

The **sources** are the representation of starting points of a search such as commercially available search-engines, email-servers, newsgroups or directories. All sources are defined in XML documents, which enables an easy maintenance and extension of the sources in a standardised format.

The **Web-API** is a set of functions and methods, which supports an easy access to the documents of the Web. The crawling agents use the Web-API for searching and crawling for Web documents, downloading them, and putting them into the document management component.

The **document management** component is responsible for the management of all documents and their metadata. It is the central component of our architecture and implements the classes and methods for the other components.

For every document the document management calls the **semantic analysis** (developed in C++) via a COM wrapper to get a relevance value for the specific document. The semantic analysis uses a semantic net that models the real world by a controlled vocabulary. This semantic net can be individually adapted to various application domains (e.g. building and construction; computer industry) to meet business demands. It consists of concepts (nodes) describing the semantics of the system by using typical relationships (typed links), like “is-a”, “consists-of” and so on. With the help of this semantic net it is possible to find also documents, which do not contain the terms of the query, but for instance a synonym, an acronym, or broader or narrower terms (see chapter 5). Another

advantage is that the results may be more precise than results from other information seeking system as homonyms can be avoided. For instance a search for "*bank*" could result in the financial institute, it could be the memory *bank* in the computer, or the *bank* at the shore. Specifying a domain-specific semantic net e.g. for the computer industry, INSYDER can determine that *bank* must have to do with the hardware of a computer and will apply this when calculating the ranking value for documents found.

The document-management component stores all retrieved web documents (only the text) in the **file-system** while the corresponding metadata are stored in the **repository** with a link to the web document in the file system. The communication between the document management component and the repository is made via the JDBC/ODBC interface.

3 Overview of the novelty of the INSYDER approach

The retrieval aspects of the visual information seeking system INSYDER have not been in the primary research focus. Nevertheless the system offers some retrieval features that are not very common in today's business intelligence tools used for Web searching.

INSYDER uses a *dynamic search* approach. The idea is to use an online search to discover relevant information by following links. The main advantage is that the current structure of the Web is searched, and not the index of a search engine. The dynamic search is based on special crawling agents. They use different heterogeneous sources (like search engines, Web directories, Web sites, documents) as starting points for the link traversal. For example the query terms are submitted to pre-selected search engines (like Altavista, Google etc.) and the hyperlinks in the search results are then used for the further crawling in the Web. This way INSYDER can also be seen as a meta search engine. All documents found are downloaded, analysed incrementally to find out how good these documents match the query. This way we can guarantee that the documents presented in the result views are up-to-date. Inquirus [Lawrence, Giles 1998] does also an online analysis and an own and therefore consistent ranking of documents found by search engines, but is designed to be a mere meta search engine, as it does not any further crawling starting with the documents found.

At the moment, our system has the drawback that we do not lookup the query in the own INSYDER repository. This means that it takes some time until search results are available, though they might be available in the own database. We know about this drawback and will do work on it. Still the main advantage of the system is the dynamic search. Unlike commercial search systems we do not intend to crawl all the WWW and store its contents, but only dedicated parts, which seem to be relevant for a given query by a user. By this way of specialising the search by specialising the crawling, e.g. for a specific branch like building and construction, we intend to increase the precision and recall compared to other meta search engines, which only rely on the results from the search engines indices.

Another special feature of INSYDER is that the relevance ranking uses a semantic analysis of documents, which is based on a *semantic net* provided with the system. Like a thesaurus the semantic net can be seen as a controlled vocabulary for the documents and the query. It offers important advantages such as identification of search terms with a clear semantic meaning, or retrieval based on concepts rather than on words. There is still an open debate how useful a thesaurus-based retrieval might be in the context of the Web, because a well known body of knowledge which can be associated with the documents in the Web does not exist. This was one of the reasons why the semantic nets used in INSYDER have been developed for two specific application domains (e.g. Building and Construction, CAD). By doing this we can also meet the demand by the business users from a Business Intelligence System to get more precise results than from a general search-engine.

As the INSYDER system has no vector model built up from a known document source (as e.g. done by search engines using their database), other ways were needed to provide common Information Retrieval functions, as e.g. query expansion and as a special case of this the relevance feedback. Therefore INSYDER uses the above mentioned semantic net. As the WWW can not be compared to a given corpus, which can be determined by a distinct number of descriptors, a new way had to be found to describe documents from the WWW efficiently in a common way, e.g. for the relevance feedback. That is why we use the concepts of the semantic net. Using the concepts we take into account the disadvantage that it is possible that the concepts proposed are to broad for a query, e.g. information is expanded to 'wire of news', 'electronic information',

'media'. Therefore we present the user in the Interactive Relevance Feedback a suggestion which terms to use.

Using a weighted query based on a semantic net is new in the field of WWW, too. Though this part has not been evaluated yet, a first impression we have is that the results tend to be more precise. Still here is also the disadvantage that this is only true for concepts being well represented in the semantic net.

An important aspect of INSYDER thinking of its novelty is the fact that ideas and components from different fields were combined. It is for sure not new to combine visualisations and information retrieval aspects, but nowadays systems which do a dynamic search with a metadata generation and the different visualisations of this metadata and document inherent data are new. Our approach aimed at getting the biggest added-value for the user combining components like dynamic search, visualisation of the query and different visualisations of the results (e.g. Scatterplot, Bargraph, Stacked Column) and information retrieval techniques (e.g. query expansion, ranking of results).

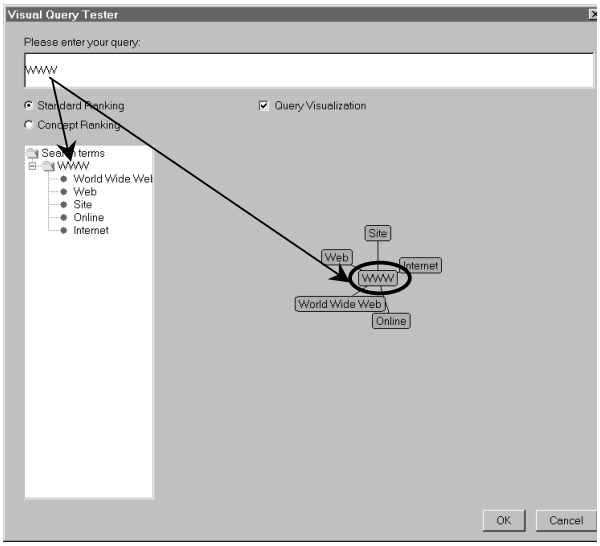

Figure 2: Prototype View of the Visual Query

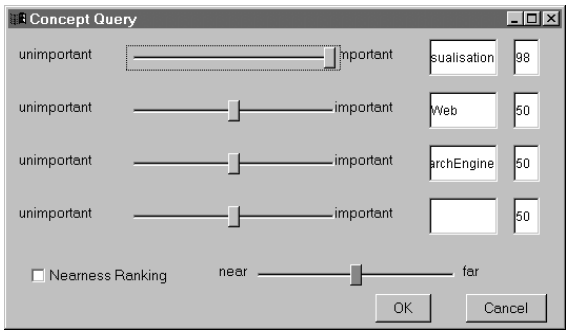

Figure 3: Dialog window for the Concept Query

4 Enhancements to the INSYDER system

4.1 Visual Query

The idea of the query visualisation is to help users to specify their information need more precisely using query expansion techniques and visualisation. From the literature it is well known that the average query consists of one or two query terms. This led to the demand of methods to overcome the problem of lacking knowledge to formulate queries. We propose therefore a visual query, which will show the user terms and relationship of terms for his query (see figure 2), taking into account other successful solutions and ideas from automatic query expansion and query visualisation, e.g. [Smeaton, Crimmins 1997], [Voorhees 1994], [Voorhees, Harmann 1998], [Zizi, Beaudouin-Lafon 1994] and [WebVibe].

First thing for the user to do is to type in his information need in a standard text entry field. This text entry field has a wider than average size as proposed by [Shneiderman, Byrd, Croft 1997] to motivate the user to type in as many terms as he can think of. In a next step this query is visualised both in a tree view and optionally in a network graph view. The query terms the user entered are the basis for the tree view (each term is a branch, unless terms occur twice). This tree is then extended by related terms, which are inserted as leafs into the tree. These related terms are taken from a semantic net, which is a basic component of the semantic analysis of the overall system INSYDER. The semantic net models the real world by representing it with concepts. For example, the WWW is defined as a concept, Web is handled as a synonym and terms like Internet, Online, Site etc. are modelled as having a strong relationship to the term WWW. If the user searches for the term WWW, the system could propose him also the terms Internet, Online, Site etc. in the Visual Query to include these in his search. At the moment, such an inclusion is made implicitly as related terms are taken into account when calculating the quality of a document (a value for the match of a query and a document found, see section 4.2). But here the related terms are only taken partly into account, not fully (unless the term is a synonym). So it would be nice for the user to have two things: first the certainty about the query and second the possibility to include more terms (the user may not have been thinking of before) into his query. The visualisation of the query has been designed taking into account several aspects:

- *The semantic net includes various relationships (e.g. part-of, broader term, narrower term):* These are not represented in the graph visualisation, just the fact that there is a relationship.
- *The user may have many input terms at first sight:* To keep the overview the system was designed having a detailed and a full view. This is simply done by taking the information from the tree view. E.g. if the user clicks on a branch of the tree view, only this branch is visualised in the graph, while clicking on the root of the tree (search terms) will result in a graphical presentation of the whole tree in the graph, which may be not easy to overview.
- *Interaction with the graph representation:* all terms represented in the graph representation can be moved keeping the relation to their base node. The elements are ordered automatically using an algorithm to make sure that when having many elements connected to a node most of them are viewable (see figure 2).

The Visual Query has been implemented prototypically. What is still missing is the connection to the semantic net to show the user directly related terms - the real added value of this feature. This problem derives from the fact that the INSYDER system brings together two programming worlds: on the one hand there is C++ for the handling of the semantic net, the semantic analysis, parts of the Relevance Feedback, and the ranking algorithms. On the other hand we have Java for the implementation of the GUI, for the assistants for the classification and visualisations and also for the Visual Query. The two worlds are communicating using a COM interface. As the access to the semantic net by COM objects has not been implemented yet and obviously appears not to be easy, this part is still missing. Meanwhile, we have thought about a workaround using a different semantic net, e.g. WordNet, but still further investigations in this direction have to be made.

4.2 Ranking

In the INSYDER system we have implemented two ranking algorithms, called Natural Language and Concept Query.

4.2.1 Natural Language

The Natural Language Algorithm is the default ranking algorithm of the system. Here, the user types in a query in a similar way he would express his information need. E.g. 'How many inhabitants has London?' This natural language query is then translated into a query for the crawler agents. Whereby translation means

the elimination of stop words (common words in the different languages, e.g. 'and', 'or', 'many' etc.) and the elimination of punctuation. Doing this the given query would lead to a final query consisting of the following terms: 'inhabitants London'. In this step we believe to have a great added value from the visual query, because the user can see which terms are finally used for searching. This way it should be much clearer for him to derive why some documents were found. The two meaningful terms extracted from the original query are then sent to different search engines (chosen by the user), to get a first result from their indexes (for an overview on crawling agents see e.g. [Turau 1998]). From these results the crawling agents excerpt the links, which are then going to be the base for further crawling. The results coming from the crawling agents are given to the ranking assistant, which calculates a number, describing how well a document found by the crawling agent matches the query. Numbers range from 0 to 100, whereby 0 means no match at all and 100 means best match possible. Documents above a ranking value of 75 are very good documents thinking in terms of satisfaction from the user's point of view. 75 means that at least in one part of the document all query terms occur. These ranking values are calculated using a semantic analysis algorithm in the background. Documents found are represented in a meta description, consisting of information about segments (which are usually similar to the sentences) and their description in the semantic net. The query itself is also put in such a meta description. This way, the system has a basis for the comparison of the query and the document. The matching itself is done segment by segment. That is to say that a value for each segment is achieved hereby. We call this set of values the detailed rank. The values of this detailed rank are used for the visualisations, e.g. the relevance curve in the result table or the various representations in the segment view. The final ranking value is then calculated, taking into account the mean value of all segments and the maximal value reached of one or more segments. The maximum value gets a higher weight in the final calculation than the mean, to make sure that documents containing all query terms (from the system generated query) in the same segment are ranked higher than documents containing the query terms in a loosely order spread all over the document. This way, to calculate the overall relevance of a document can be compared to a Boolean 'AND' with a 'NEAR' proximity operator. The crawling agents will also find documents containing only some terms of the original query, having a much smaller value for the overall ranking value. This means an 'OR' compared to systems using Boolean logic. At the moment, there is no indication for this for the user, but this is planned for a future version of the system, e.g. as a representation in a scale.

Hereby studies saying that the Boolean logic is too complex to most users shall be taken into account when designing this scale. [Shneiderman, Byrd, Croft 1997] for instance, propose to use a verbal description, e.g. 'contain all words', 'contain one or more words', and so forth.

4.2.2 Concept Query

The Concept Query has been designed to give the user a higher level of influence and interaction, defining his information need. This Concept Query is not seen as a possibility for beginners in Web searching, but for advanced users. The idea behind is that the user himself defines how important the different query terms are for the satisfaction of his information need. Therefore the user weights the different concepts of his query. Speaking of query we mean a concept which must be defined in the semantic net. In figure 3 a prototype version of the implementation can be seen. The user asked here for the concepts 'Visualisation', 'Web' and 'SearchEngine'. Obviously the user is very much interested in the concept of 'Visualisation', giving this concept an 'important' value of 98, while the other two concepts are not seen as important, both given a weight of 50. In the dialog window a nearness check box with a slider can be seen as well. This is another interaction possibility for the user. Here the user has the possibility to define, if all the concepts should appear next to each other (*near* e.g. in the same segment) or could be spread all over the document (*far*). Depending on the selection the final ranking value for the documents differs. As in the *Natural Language* algorithm, the overall ranking value is derived from the detailed rank value, calculated for each segment.

For the moment, the use of the Concept Query is rather uncomfortable, as the user has to know the concept terms from the semantic net. This means there is temporarily no translation from a given query (which could be natural language) to the concepts. In the example above this would be no problem, as 'Inhabitant' and 'London' are both concepts of the semantic net.

At this point it is planned to combine the Concept Query window with the Visual Query window. Instead of having a slider for the weight, we think of having a spin button next to the concepts in the graph or where the user can define the weight then. Another idea we have in mind is the use of colour and thickness of the concepts, e.g. the user could set the weight by a slider coming up when clicking with the right mouse button or by a separate dialog box. The intensity of the concept itself (or the frame) could then be a visual indicator for the weight of

a concept (fade - unimportant, strong - important). These ideas have been thought about, but there is no prototypical implementation available yet.

4.2.3 Evaluation planned

The effectiveness of the system shall be measured by using the *tfidf* (term frequency inverse document frequency) measurement as a baseline for the ranking algorithms. Therefore the tfidf measurement has been integrated into the system. The basis for the *idf* measurement is the actual document collection, meaning the documents which have been found by the web crawling agents. The tfidf is calculated taking the formula of [Salton, McGill 1987]. Thinking about measuring the precision at different cut-off levels, we take the studies of [Buckley, Voorhess 2000] into account. They suggest that a cut-off level of 10 to 20 makes sense for the evaluation of Web search engines, but also point out that with increasing number of topics (queries to the search systems) the error level decreases. For our evaluation purpose we have therefore decided to choose 30 of the topics provided by [TREC]. The topics are described by keywords, phrases and give a narrative, what a document has to fulfil to be judged relevant. It is planned to measure the precision at a cut-off level of 20 as maximum. This is done to keep the error level at a low value and still to be able to handle the evaluation data. It is also planned to do an evaluation of different systems, similar to the evaluation of search engines by [Leighton, Srivastava 1997]. The evaluation will differ from their evaluation, as meta search engines and local search engines are going to be used. The purpose of this evaluation will mainly focus on the efficiency of the systems finding relevant information. For the time being we are working on the test settings.

4.3 Interactive Relevance Feedback

As it has been said before INSYDERs retrieval functions base very much on the thesaurus. Therefore it was decided to give the user just suggestions for terms for the relevance feedback as otherwise the documents found (using a fully automatic relevance feedback) would be too broad (low in precision), covering too many different subjects. Hence INSYDER offers the function of an Interactive Relevance Feedback, meaning the use of judgements made by the user about documents (like, dislike) to derive from these new query terms, which are suggested to the user. The first approach was to extract the feature concepts of a document. Meaning that an analysis of the positive or negative judged document is made, extracting from each document those concepts describing it

best. Hereby best describing means that this concept occurs most often in a document. As the concept 'proper name' might occur several times derived from different nouns (e.g. Smith, Hull, Clarke would be three times the concept proper name), these concepts are ignored, the same applies for concepts like number etc. and verbs. The remaining concepts are ordered according to their frequency. The frequency itself is calculated using a positive and a negative weight. Concepts from documents judged as relevant ('find more like this') are added while negative concepts are subtracted. In the example about a search on "information visualisation" (see figure 4) we can see that the user decided to have more like the first two documents and nothing similar to the third one.

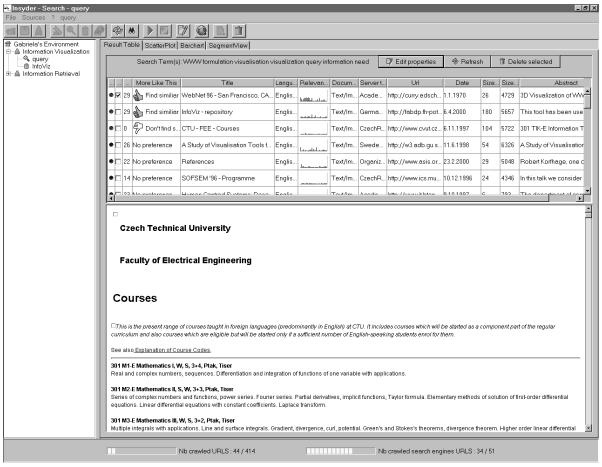

Figure 4: View of the INSYDER system with relevance judgement by the user

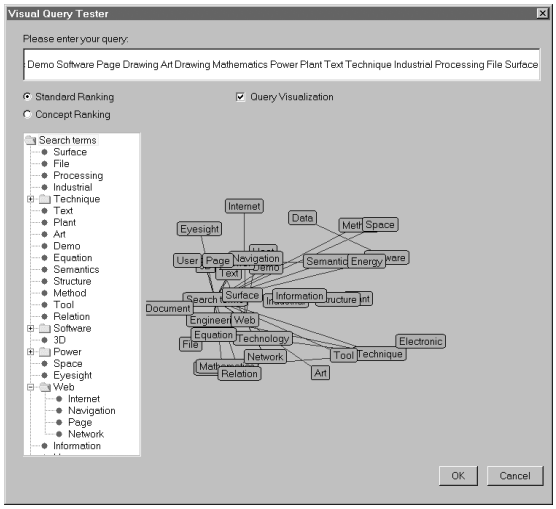

Figure 5: Prototype View of the Visual Query with an overview presentation of the input from the Relevance Feedback

A negative thing about the fact that proper names are eliminated is that if a document e.g. is about the biography of a special person, the person's name will never occur in the Relevance Feedback query. As it was not possible for technical reasons to get the real name of the concepts (e.g. instead of proper name - Smith), we decided to try the classical approach of text analysis, too. Therefore we use the Porter algorithm for English documents [Porter 1997] beside the elimination of stopwords to get the feature terms of the documents. Using the Porter Algorithm we get the principle forms of all the words in the document. By calculating their frequency and the re-translation to the original term, we get the possibility to include proper names in the Relevance Feedback

query, too. We decided to use a mixture of these two algorithms to get the best out of the two of them: by the concepts the user widens his view of things, by using the classical approach we do not have to sacrifice the proper names. Still we found out that the use of the two algorithms is very time consuming (particularly the classical algorithm), so that we are thinking about providing only the concept algorithm instead of the hybrid version of both of them, giving the user the option of using both of them, if he wants (e.g. for a second refinement step). The query terms created are presented to the user in the Visual Query window (see figure 5). The user has the possibility to have an overview or detailed view of the feature concepts and terms as described in chapter 5. Unfortunately, here are also the relations from the semantic net missing, as for technical reasons there is no direct connection available yet. From the Visual Query the user can select terms to go on with.

5 Conclusion and Outlook

In this paper we have focused on the description of enhancements to the existing Information Assistant INSYDER. These were in particular the Visual Query and the Concept Query Ranking Algorithm. Apart from these the Natural Language Ranking Algorithm and the enhancement of the systems feature the Relevance Feedback using concepts was explained. Some of the ideas presented are still ideas and need to be implemented. Throughout the ideas presented above we are still working on the enhancement of the overall system. This includes the visualisations of the search results, the visualisation algorithms and particularly the user interface of the whole application. We have started to redesign the system using the Java Web Style Guide [Sun 2000]. Still we are working to solve the problem to get a direct connection to the semantic net in close co-operation with our French partner ARISEM. Moreover evaluations of the ranking, the Relevance Feedback and the usability of the Visual Query have to be made soon.

Acknowledgements

The authors wish to thank the following EU-ESPRIT project partners for their contributions and support: Alain Garnier, Olivier Spinelli, Laurent Dosdat (ARISEM, Paris); Guillaume Lory, Carlo Revelli (Cybion, Paris); Rina Angeletti (Innova, Rome); Flavia D'Auria (Promoroma, Rome).

References

[Arisem] http://www.arisem.com [2000-01-18].

[Buckley, Voorhees 2000] C. Buckley, E. Voorhees. Evaluating Evaluation Measure Stability. Proceedings of the annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '00, Athens 24-28 July 2000.

[Hackathorn 1998] R. Hackathorn. Web farming for the data warehouse. Morgan Kaufmann, San Francisco, 1998 .

[IBM] http://www.software.ibm.com/data/pubs/papers/bisolution/index.html [2000-01-18].

[Koenemann, Belkin 1999] J. Koenemann and N. J. Belkin. A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. CHI 96 - Electronic Proceedings. R. Bilger, S. Guest and M. J. Tauber (eds.). http://www.uni-paderborn.de/StaffWeb/chi96/ElPub/WWW/chi96www/papers/Koenemann/jk1_txt.htm [1999-11-11].

[Kuhlen 1999] R. Kuhlen. Die Konsequenzen von Informationsassistenten. Was bedeutet informationelle Autonomie oder wie kann Vertrauen in elektronische Dienste in offenen Informationsmärkten gesichert werden? Frankfurt (Suhrkamp-Verlag). 1999.

[Lawrence, Giles 1998] S. Lawrence and L. Giles. Inquirus, The NECI Meta Search Engine, Proceedings of the Seventh International World Wide Web Conference, Brisbane, Australia, 1998.

[Leighton, Srivastava 1997] V. Leighton; J. Srivastava: Precision among World Wide Web Search Services (search engines). http://www.winona.msus.edu/library/webind2/webind2.htm [1999-03-02]

[Mann, Reiterer 1999] T. M. Mann, H. Reiterer. Case Study: A Combined Visualization Approach for WWW-Search Results. IEEE Information Visualization Symposium. N. Gershon, J. Dill and G. Wills (eds.). 1999 Late Breaking Hot Topics Proc. Supplement to: G. Will, D. Keim (eds.): Proc. 1999 IEEE Symposium on Information Visualization (InfoVis'99). Conference: San Francisco, CA, USA, October 24-29, 1999. Los Alamitos, CA (IEEE Computer Soc. Press). San Francisco 1999: 59-62. 1999.

[Mann, Reiterer 2000] T. M. Mann, H. Reiterer. Evaluation of different Visualizations of WWW Search Results. Proc. Eleventh International Workshops on Database and Expert Systems Applications (DEXA 2000). Conference: Greenwich, UK, September 4-8, 2000 (IEEE Computer Society).

[Mußler 1999] G. Mußler. Ein Agentensystem zur Unterstützung bei der Informationssuche im WWW. Electronic Proceedings http://wwwdb.informatik.uni-rostock.de/adi99/endversionGM.ps [1999-10-05].

[Nielsen 1997] J. Nielsen. Search and You May Find. http://www.useit.com/alertbox/9707b.html [1999-03-18].

[Pollock, Hockley 1997] A. Pollock and A. Hockley. What's Wrong with Internet Searching. D-Lib Magazine, 1997, http://www.dlib.org/ dlib/march97/bt/03pollock.html [1999-02-01].

[Porter 1997] M. Porter. An algorithm for suffix stripping. Readings in Information Retrieval. K. Sparck-Jones and P. Willett (eds.). Morgan Kaufmann, San Francisco, 1997.

[Reiterer, Mussler, Mann et al. 2000] H. Reiterer, G. Mußler, T. M. Mann and S. Handschuh: INSYDER - An Information Assistant for Business Intelligence. Proceedings of the annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '00, Athens 24-28 July 2000.

[Salton, McGill 1987] G. Salton and M. J. McGill. Information Retrieval - Grundlegendes für Informationswissenschaftler. McGraw-Hill, 1987.

[Shneiderman, Byrd, Croft 1997] B. Shneiderman, D. Byrd, and W. B. Croft. Clarifying Search: A User-Interface Framework for Text Searches. D-Lib Magazine, 1997, http://www.dlib.org/dlib/january97/retrieval/01shneiderman.html [1999-08-17].

[Smeaton, Crimmins 1997] A. F. Smeaton and F. Crimmins. Relevance Feedback and Query Expansion for Searching the Web: A Model for Searching a Digital Library. Research and Advanced Technology for Digital Libraries. P. C. Pisa and C. Thanos (eds.). First European Conference, ECDL'97 (Springer) 1997.

[SUN 2000] http://java.sun.com/products/jlf/dg/higtitle.alt.htm [2000-09-04].

[TREC] TREC-8 ad hoc and small web topics http://trec.nist.gov/data/topics_eng/topics.401-450.gz [2000-08-24].

[Turau 1998] V. Turau. Web-Roboter. Informatik Spektrum, 21(3): 159-160, 1998.

[Voorhees 1994] E. M. Voorhees. Query Expansion using Lexical-Semantic Relations. Proceedings of the seventeenth annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '94, 3-6 July 1994. B. W. Croft, (eds.). Dublin, Ireland, London, Berlin u.a. (Springer): 61-69, 1994.

[Voorhees, Harman 1998] E. M. Voorhees and D. K. Harman (eds.): NIST Special Publication 500-242: The Seventh Text Retrieval Conference (TREC-7) Gaithersburg, Maryland (Government Printing Office (GPO)) 1998. http://trec.nist.gov/pubs/trec7/t7_proceedings.html [1999-12-20].

[WebVibe] http://www2.sis.pitt.edu/~webvibe/WebVibe/ [2000-09-06].

[Zamir, Etzioni 1998] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. SIGIR 1998. http://zhadum.cs. washington.edu/zamir/sigir98.ps [1999-03-23].

[Zizi; Beaudouin-Lafon 1994] M. Zizi; M. Beaudouin-Lafon. Accessing Hyperdocuments through Interactive Dynamic Maps. Conference on Hypertext and Hypermedia Proceedings of the 1994 ACM European conference on Hypermedia technology. 126-134.