Int J Digit Libr (2005) / Digital Object Identifier (DOI) 10.1007/s00799-004-0111-y

INSYDER: a content-based visual-information-seeking system for the Web

Harald Reiterer[1], **Gabriela Tullius**[2], **Thomas M. Mann**[3]

[1] Computer & Information Science, University of Konstanz, 78457 Konstanz, Germany
e-mail: harald.reiterer@uni-konstanz.de
[2] School of Informatics, Reutlingen University, 72762 Reutlingen, Germany
e-mail: gabriela.tullius@fh-reutlingen.de
[3] e-mail: Thomas.Mann@t-online.de

Published online: 3 March 2005 – © Springer-Verlag 2005

Abstract. This paper presents INSYDER, a content-based visual-information-seeking system for the Web. The Web can be seen as one huge digital library offering a variety of very useful information for business analysts. INSYDER addresses these possibilities and offers powerful retrieval and visualisation functionalities. The main focus during the development was on the usability of the system. Therefore, a variety of well-established visualisation components were employed to support the user during the information-seeking process (e.g. visual query, result table, bar graph, segment view with tile bars, and scatterplot). Also, the retrieval aspects were developed with the goal of increasing the usability of the system (e.g. natural language search, content-based classification, relevance feedback). Extensive evaluations of the retrieval performance and the usability of the visualisation were conducted. The results of these evaluations offered many helpful insights into developing a new visual-information-seeking system called VisMeB.

Keywords: Information visualisation – Information seeking – Usability

1 Introduction

INSYDER (INternet SYstem DE Recherche),[1] a visual-information-seeking system, was developed with the objective of finding, analysing, and monitoring business information on the Web and presenting it in an intuitive way.

Current search systems lack usability, e.g. users have difficulties using search facilities and are not satisfied with the search results [7, 23, 39, 42]. With INSYDER we propose a way to overcome these problems. Users are provided with an added-value system characterised by an overall improved usability, and the usability is enhanced by providing users with a content-based system, a different retrieval strategy, and a new way of presenting search results.

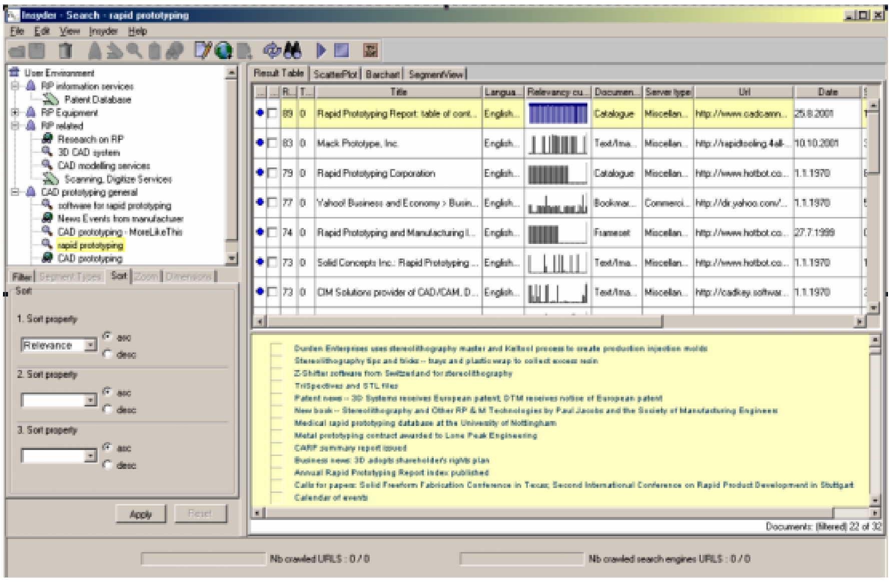
This paper will first explain the idea behind INSYDER, then show the visual-information-seeking support, and finish with an evaluation of the retrieval performance and of the usability of the search result visualisation.

2 INSYDER – a content-based system

The purpose of INSYDER is not to act as another search engine, e.g. AltaVista or Google, but to be a content-based search assistant. This new way of designing a search assistant means that the user first of all gets a preconfigured system that meets his needs. The basic design of INSYDER is to give the user a sphere of interest (SOI), where he can organise all the information concerning his information needs: searches (marked with a magnifier 🔍), watches (marked with binoculars 🔭), news (marked with a notepad 📋), and bookmarks (marked with a bookmark 🔖). The organisation of the SOI is subject-based; a sphere expresses a user's information need, which might be manifold. For example, if the user is a business manager in the sales department, one SOI could be named *customers*, including searches and watches about customers and bookmarks to their Web presentations. Another sphere could be *competitors*, e.g. including searches for their products and the customers they serve. Figure 1 shows a screen shot of the INSYDER GUI.

The INSYDER search mechanism uses its own knowledge base (KB), a kind of thesaurus. This KB provides a general view of the world, but it can be adapted to the user's needs, e.g. by classifying the business focus in depth

[1] This project was funded by the European Union (Esprit project #29232).


Insyder - Search - rapid prototyping
File Edit View Insyder Help
Result Table | ScatterPlot | Barchart | SegmentView
Filter | Segment Types | Sort | Zoom | Dimensions
Apply
Reset

User Environment
- RP information services
  - Patent Database
- RP Equipment
- RP related
  - Research on RP
  - 3D CAD system
  - CAD modelling services
  - Scanning, Digitize Services
- CAD prototyping general
  - software for rapid prototyping
  - News Events from manufacturer
  - CAD prototyping - MoreLikeThis
  - rapid prototyping
  - CAD prototyping

Sort

1. Sort property: Relevance ● asc ○ desc

2. Sort property: ● asc ○ desc

3. Sort property: ● asc ○ desc

...	...	R...	T...	Title	Langua...	Relevancy cu...	Documen...	Server type	Url	Date
●	□	89	0	Rapid Prototyping Report: table of cont...	English...		Catalogue	Miscellan...	http://www.cadcamn...	25.8.2001
●	□	83	0	Mack Prototype, Inc	English...		Text/Ima...	Miscellan...	http://rapidtooling.4all-...	10.10.2001
●	□	79	0	Rapid Prototyping Corporation	English...		Catalogue	Miscellan...	http://www.hotbot.co...	1.1.1970
●	□	77	0	Yahoo! Business and Economy > Busin...	English...		Bookmar...	Commerci...	http://dir.yahoo.com/...	1.1.1970
●	□	74	0	Rapid Prototyping and Manufacturing I...	English...		Frameset	Miscellan...	http://www.hotbot.co...	27.7.1999
●	□	73	0	Solid Concepts Inc.: Rapid Prototyping ...	English...		Text/Ima...	Miscellan...	http://www.hotbot.co...	1.1.1970
●	□	73	0	CIM Solutions provider of CAD/CAM, D...	English...		Text/Ima...	Miscellan...	http://cadkey.softwar...	1.1.1970

Durden Enterprises uses stereolithography master and Keltool process to create production injection molds
Stereolithography tips and tricks -- trays and plastic wrap to collect excess resin
Z-Shifter software from Switzerland for stereolithography
TriSpectives and STL files
Patent news -- 3D Systems receives European patent; DTM receives notice of European patent
New book -- Stereolithography and Other RP & M Technologies by Paul Jacobs and the Society of Manufacturing Engineers
Medical rapid prototyping database at the University of Nottingham
Metal prototyping contract awarded to Lone Peak Engineering
CARP summary report issued
Business news: 3D adopts shareholder's rights plan
Annual Rapid Prototyping Report index published
Calls for papers: Solid Freeform Fabrication Conference in Texas; Second International Conference on Rapid Product Development in Stuttgart
Calendar of events

Documents: (filtered) 22 of 32

Nb crawled URLS : 0 / 0

Nb crawled search engines URLS : 0 / 0

Fig. 1. The INSYDER GUI showing the sphere of interest on the *left*

and in different languages. It is obvious that such systems have to be administrated to keep the content information up-to-date. It was intended to keep the content administration as easy as possible, e.g. the definitions of sources are in XML.

In the different steps of a search process, the user is in each step provided with content (Fig. 2).

2.1 Outline of the system

The user of INSYDER is provided with comprehensive visual support, so INSYDER [34, 38] can be seen as a visual-information-seeking system [46]. The visual support is found in various phases of the search process (Fig. 2). The four-phase framework of Shneiderman et al. [50, 51] has been chosen as it provides a practical, though sophisticated enough, task model for the different phases of information seeking [45]. The formulation phase is supported by using the visual query to express the information need and a wizard-like dialogue for the selection of the sources. For a review of results, meta-information is generated. Its visualisation is the focus of the third phase in the framework. Also in this phase the user can apply filters to get results that only match certain attributes (e.g. filter documents from commercial sites etc.). For the formulation of the next step in the refinement phase, a relevance feedback option has been implemented that links the last to the first (formulation) phase. As for search term suggestions, visual queries are also used.

2.2 System architecture

The INSYDER system consists of several components (Fig. 3). The lower tier is the foundation of the processing tier: the KB, the source definition for the searches, and the server-type definitions for the classification part. Microsoft SQL Server is used to store the meta-information of the search results. The documents themselves are stored using the operating system's flat file structure.

Apart from the *semantic analysis* component, all components are implemented using Java. The semantic analysis component is based on existing C++ modules from the project partner ARISEM (`http://www.arisem.com`, retrieved 10 Oct 2003), which could be reused and en-

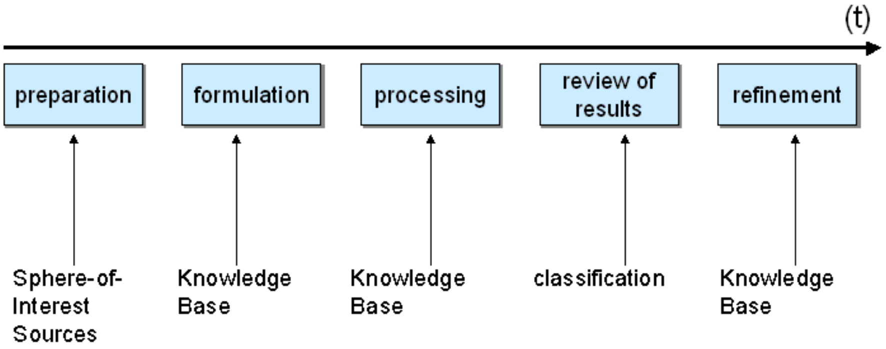


Fig. 2. Content provision in the search process

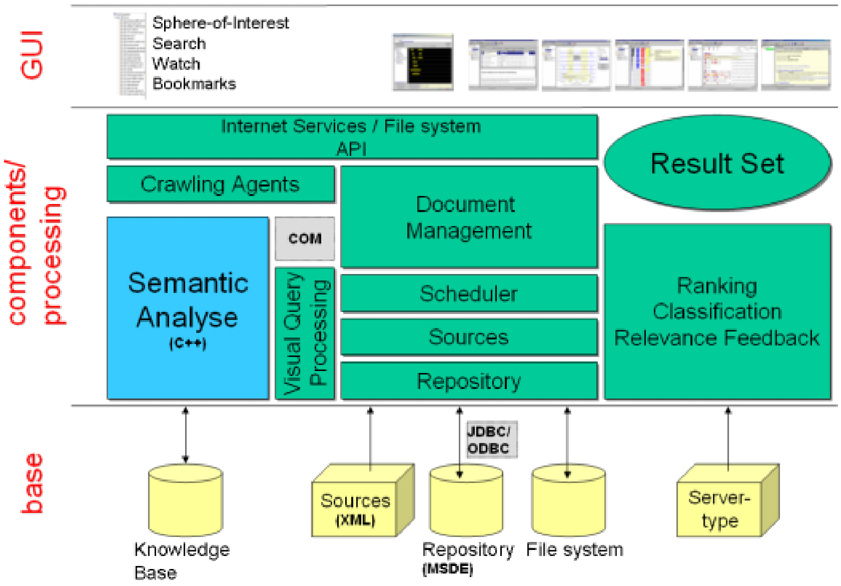


Fig. 3. INSYDER architecture

hanced for the INSYDER project. The *document management* component is the central component of the system. This component makes all documents accessible in all stages of their processing and triggers further actions (calculation of metadata). The input for the system is either documents from Internet services like the Web or the local file system from the user's PC or network. Naturally both have their own implementation of an *API*. For the search on the Web, *crawling agents* are necessary, which use the hypertext structure of the Web, following the links of a distinct page and returning the new pages found to the document management component. The *ranking, classification*, and *relevance feedback* components analyse and classify the documents found. The *scheduler* is necessary when performing a monitoring of Web sites to trigger further events. The *sources* and *repository* components are the counterparts for the base components, processing the input (i.e. sources) and communicating with the database and the file system, respectively. Figure 3 shows that the *result set* component is drawn in a circle; this is to make it clear that the result set exists as an abstract construct, being substantiated by the different variations of the result visualisations. The GUI tier is responsible for the different result visualisations and the visual query.

3 Information visualisation

The human perceptual system is highly adjusted to process visual coded information very effectively. In the last few decades such visualisations using computers have developed into an independent technical discipline within the area of human-computer interaction. Information visualisation (IV) in INSYDER is used for the formulation phase of a search, for the review of search results, and for refinement. For all the visualisations a rather simple approach has been chosen [33].

3.1 Formulation phase

Research shows that users have problems formulating their information needs [7, 23, 39, 42]. Query expansion in particular seems to be a problem for novice users [54]. This has led to the demand for methods to overcome the problem of lack of knowledge needed to formulate queries. The idea of the visual query formulation is to help users specify their information needs more precisely using interactive query expansion techniques based upon visualisation.

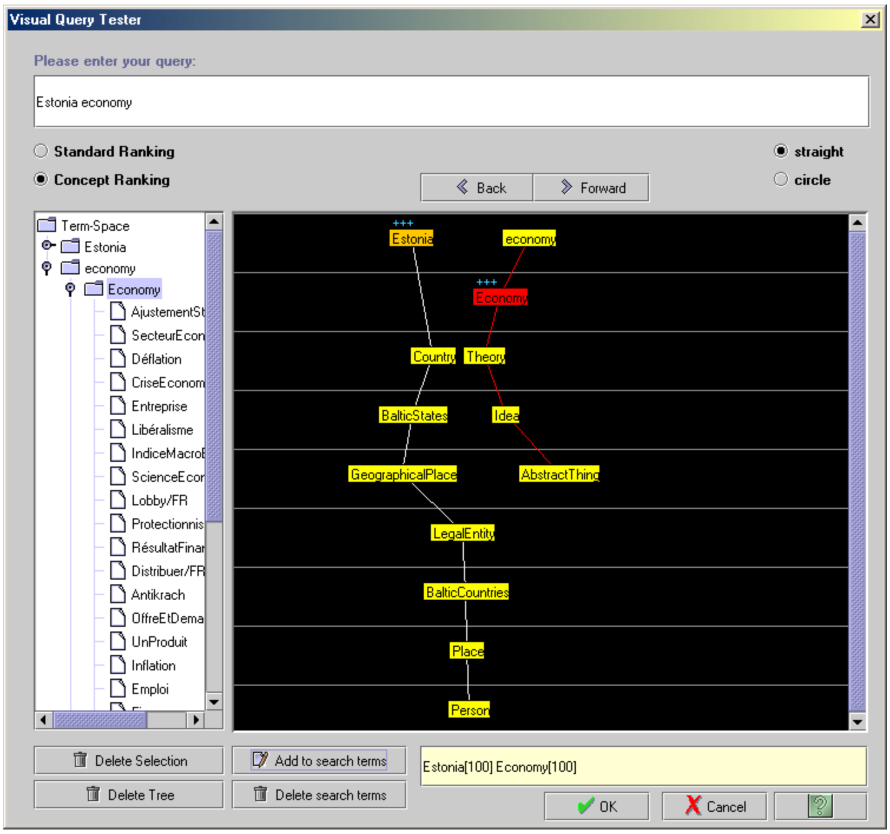
Within the literature little attention has been paid to the visualisation of the interactive query expansion, though the user interface is crucial for motivating users to select appropriate expansion terms and is therefore a key factor in the actual effectiveness of the retrieval [6]. Mostly the expansion terms are presented in a list-like user interface, as for example in the *Okapi* system [6].

A number of visualisations for query formulation are known in the field of IV. Strongly related to query formulation are research topics dealing with the browsing of large collections, i.e. clustering them, navigating within, and retrieving information. A popular approach to clustering document collections onto meaningful maps for exploration and search is the use of SOMs (self-organising maps) introduced by Kohonen. A special SOM architecture called *WEBSOM* has been developed for exploratory textual data mining [28]. In a common interaction model for browsing large collections such as the Web, the user provides some terms as starting points; the collection is then displayed under this point of view [1]. *GUIDO* and *VIBE* are examples of such systems. The user defines a point of interest, and the system then positions the icons (as a representation for the documents) depending on their relationship to the point of interest [29]. *InfoCrystal* [53] and *VQuery* [24] use the idea of Venn diagrams to support query formulation. Eibl [13] presents a query visualisation system called *DEViD* that combines IV and graphic design aspects, enhancing the idea of the InfoCrystal. Zizi and Beaudouin-Lafon [59] present the idea of *interactive dynamic maps* (IDMs) for the exploration of Web documents and their semantic content. Other examples are the use of 3D visualisation (e.g. *ConeTrees* [47], used, for example, by *Cat-a-Cone* [20]), or generally the visualisation of networks using various metaphors in 2D and 3D (e.g. *SemNet* [15], *Narcissus* [21] and its further development *HyperSpace* [57], *GeoSpace* [31], *SearchVis* [36]), or the combination of *starfield* displays [1] and *moveable filters* [16].

The discussed systems all have in common that the collection is known and that therefore the outcome of the query is also known. The focus of their intention is not the expansion of the query but the formulation, mostly by showing the influences of selected terms on the result set. By contrast, the intention of the visual query is to support the user interactively, expanding the initial query. As the collection size is not known (the collection is the Web), the influence of the terms on the final result set is not obtainable. This is strongly related to the (pure) visualisation of thesauri. Traditionally thesauri list term entries and, if connected to a database, the number of times the descriptor occurs in the distinct database. The *HiBrowser* user interface (`http://www.hud.ac.uk/schools/cedar/dorking.htm`, retrieved 25 Oct 2001) is an example of such a thesaurus. The *Plumb Design Visual Thesaurus* is a visual thesaurus browser showing relations in the English language using WordNet (`http://www.visualthesaurus.com/`, retrieved 27 July 2001).Ahlberg and Shneiderman [17] present the *Information Navigator*, a visual-information-retrieval system.

3.1.1 Design of the visual query

The objective of the visual query is to provide the user with an interactive query expansion (IQE). This objective was inspired by previous works [3, 17, 18, 40] and by Refine, which is used by AltaVista. Users benefit from using the visual query in two ways: first, by browsing the KB to find more accurate query terms, leading to a more precise result set, and, secondly, in the sense of an IQE by expanding their original query with additional terms from the KB, which will result in a broadened result set, which could be much more satisfying, too, as a higher recall can be expected. As a side effect, use of terms from the KB can minimise spelling mistakes, too. If a query term is not found, a warning dialogue is shown, asking the user if he is sure about the spelling or whether the po-


Visual Query Tester

Please enter your query:

Estonia economy

○ Standard Ranking
● Concept Ranking

● straight
○ circle

Estonia[100] Economy[100]

Fig. 4. Visual query

tential false term is to be deleted. It would be desirable at this stage to suggest some similar (phonetic or homographic) terms to the user so that he can easily correct any misspelling, but this is not possible with the current system. The basic layout of the visual query screen can be seen in Fig. 4.

The visualisation of the query has been designed to take into account several aspects [38]. *The visual query will, as far as possible, also follow the visual-information-seeking mantra*: Overview first, zoom and filter, then details-on-demand. *The entry points for the visualisation are the query terms entered by the user*: The results of a small study [25] suggest using long entry fields if longer queries are desired (see also [51]). *The user may have many input terms at first sight*: To keep the overview, the system was designed with a detailed and a full view. This is done simply by taking the information from the tree view. *As the KB is built using nodes and links, an intuitive visualisation seems to be a graph network*: Using the graph metaphor a basic requirement is the minimisation of crossings to avoid ambiguity and artificial salience. *Interaction with the graph representation*: All terms represented in the graph representation can be moved, keeping the relations at their base node. The elements are ordered automatically to make sure that when many elements are connected to a node, most of them are viewable. *The KB includes various relationships (e.g. is-a, part-of, broader term, narrower term)*: These are not represented in the graph visualisation, just the fact that there is a relationship. *As within a hypertext system a history shall be kept*: The user shall have the option to move within the various graphs he has already looked at (e.g. at different detail levels). *INSYDER provides two ranking algorithms*: The ranking type is determined at query formulation time; thus the visual query has to support both.

3.2 Review of search results

We now focus on the visual representations used for the review-of-results phase of the four-phase framework. The motivation for using visualisations beyond pure presentation in list form was to improve access to the abstract result sets from Web searches. The visualisations were chosen, with an emphasis on the visualisation of the search result set, to have a multiple-coordinated-view approach and to orient on business graphics. On the result set level, an overview of all search results to identify which documents fit best with the user's information needs would be useful. On the document level, the user is interested in seeing which parts of a document fit best with his information needs.

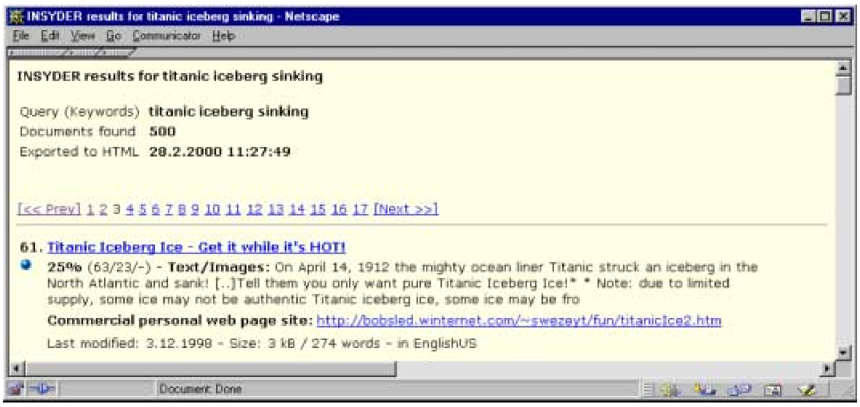
An important design decision for the result phase was to use a multiple-view approach. This is in harmony with the rule of diversity (i.e. use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction, or genres) from Baldonado et al. [5]. The visualisation of search results is a natural candidate for multiple-view approaches because of the variety of different levels of abstraction necessary to deal with search results ranging from overviews about the whole result set to detailed views of documents and their parts.

To avoid the possible drawbacks of multiple-view approaches, several guidelines were considered. The number of used visualisations has been reduced to a small number. This accords with the rule of parsimony (i.e. use multiple views minimally) from Baldonado et al. [5]. Only simple visualisations have been chosen. The visual structures have been adapted to each other based on colour, orientation, and overall style. The visualisations are synchronised in such a way that a selection in one representation of the result set will be updated immediately in the other representations too. These points are in harmony with the rules of self-evidence and consistency.

3.2.1 Design of the search result visualisations

In the early stages of INSYDER's development, a field study was conducted using a questionnaire that was answered by 73 selected companies in Italy, France, and Great Britain. The aim was to understand the context of use [22] in keeping with a human-centred design approach.[2] The typical users of the INSYDER system are experts from business domains like CAD software or building and construction. These two business domains were chosen as test areas in the project. Experts from these domains are typically not specialists in using information retrieval systems; they are familiar with the Web and have some limited understanding of search engines. The scenarios show the typical information sources, typical user information needs, and the expected functionality (search, monitoring, portal for news). These results correspond very well to an empirical study conducted by [11]. The final selection of the visual structures was based on the above suggestions of the field study, an extensive study of the state of the art in visualising text documents, and the design goal of orienting our visual structures as much as possible on typical business graphics. The field study showed that all users had a good understanding of this kind of graphics and used them during their daily work (e.g. in spreadsheet programmes). It was not the intention during the development of the INSYDER system to come up with new visual metaphors supporting the retrieval process. The main idea was to select existing visualisations for text documents and to combine them in a novel way to enhance the overall usability of a search system. We tried to select expressive visualisations, keeping in mind the target users (business analysts), their typical tasks (finding business data on the Web), their technical environment (typically a desktop PC and not a high-end workstation for extraordinary graphic representations), type of data to be visualised

[2] ISO/DIS 13 407 Human Centred Design for Interactive Systems


INSYDER results for titanic iceberg sinking - Netscape
File Edit View Go Communicator Help

INSYDER results for titanic iceberg sinking

Query (Keywords) **titanic iceberg sinking**
Documents found **500**
Exported to HTML **28.2.2000 11:27:49**

[<< Prev] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 [Next >>]

61. Titanic Iceberg Ice - Get it while it's HOT!
25% (63/23/-) - **Text/Images:** On April 14, 1912 the mighty ocean liner Titanic struck an iceberg in the North Atlantic and sank! [..]Tell them you only want pure Titanic Iceberg Ice!" * Note: due to limited supply, some ice may not be authentic Titanic iceberg ice, some ice may be fro
Commercial personal web page site: http://bobsled.winternet.com/~swezeyt/fun/titanicIce2.htm
Last modified: 3.12.1998 - Size: 3 kB / 274 words - in EnglishUS

Document: Done

Fig. 5. HTML list in INSYDER

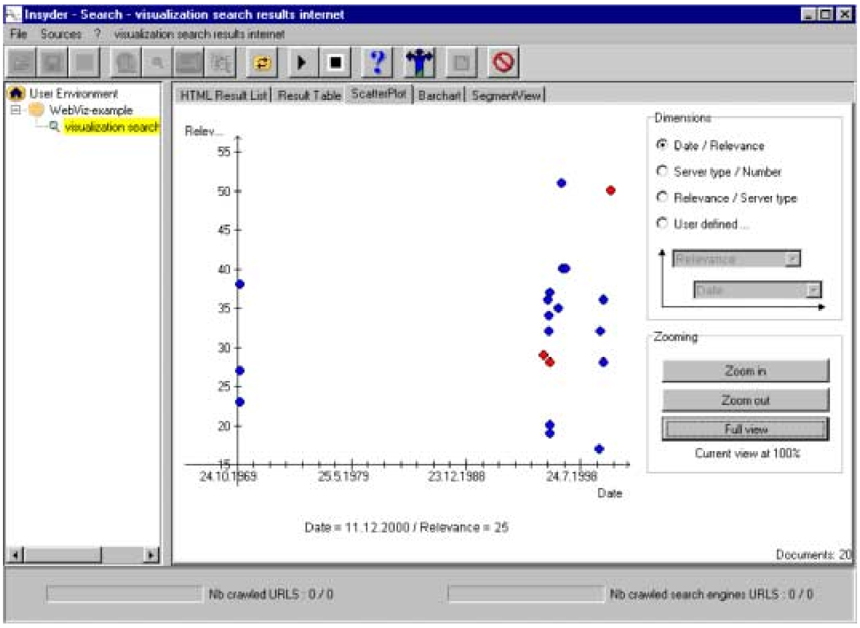
(document sets and text documents), and minimal necessary training.

3.2.2 The INSYDER visualisation components

INSYDER offers the option of showing search results in a traditional HTML format with 30 hits per page (Fig. 5). The HTML list offers the user a familiar visualisation and allows comparisons with typical presentations in common search engines. The document titles are linked to the local copy of the document, the URLs to the original document.

The second component is the result table implemented in Java (Fig. 1). All attributes from the HTML list plus a relevance curve and a relevance feedback flag are presented in table columns. The user can sort the documents by each variable in increasing or decreasing order or customise the table to his personal preferences. On the same pane as the result table, a browser is integrated that shows the locally stored version of the currently selected document.

Besides considerations about using business-graphic-like visual structures, the use of the scatterplot (Fig. 6) was inspired by visual-information-seeking systems like *FilmFinder* [1], *IVEE* [2], *Spotfire Pro* (`http://www.spotfire.com`, retrieved 22 Feb 2001), and *Envision* [41]. In the INSYDER ScatterPlot, each document is represented by a blue or red coloured dot (depending on the select status). The X and Y dimensions encode two variables. Informal tests with the prototypes revealed that it may be a good idea to guide the user by offering a small number of selected, predefined possibilities while still offering power users the possibility of choosing their own combinations. There are three predefined scatterplots available, each with a fixed definition of the X and Y (X/Y) dimensions: date/relevance, server type/number



● Date / Relevance
○ Server type / Number
○ Relevance / Server type
○ User defined ...

Current view at 100%

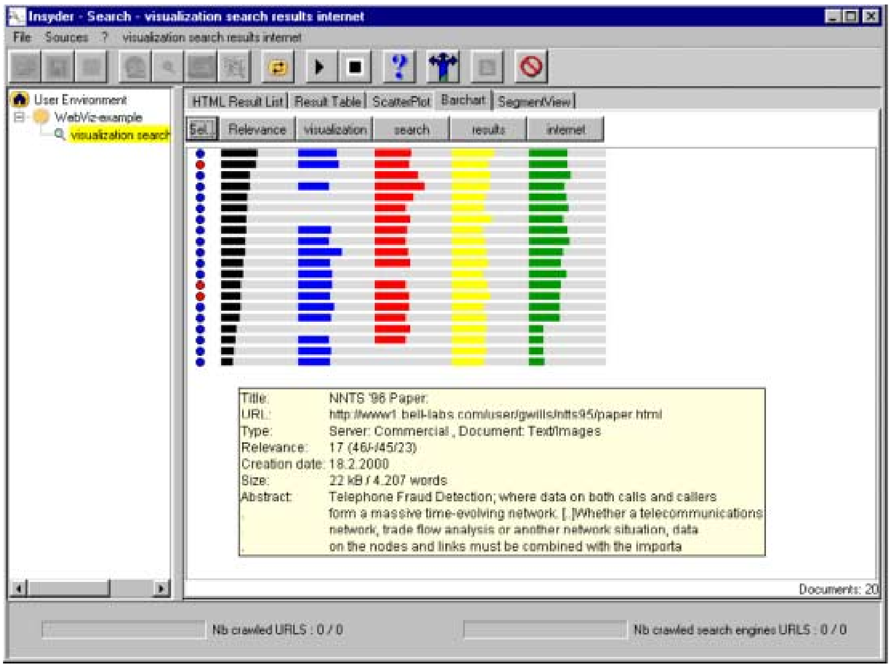
Date = 11.12.2000 / Relevance = 25

Documents: 20

Nb crawled URLS : 0 / 0

Nb crawled search engines URLS : 0 / 0

Fig. 6. Scatterplot



Title: NNTS '96 Paper:
URL: http://www1.bell-labs.com/user/gwills/nnts95/paper.html
Type: Server: Commercial , Document: Text/Images
Relevance: 17 (46/-/45/23)
Creation date: 18.2.2000
Size: 22 kB / 4.207 words
Abstract: Telephone Fraud Detection; where data on both calls and callers form a massive time-evolving network. [..]Whether a telecommunications network, trade flow analysis or another network situation, data on the nodes and links must be combined with the importa

Documents: 20

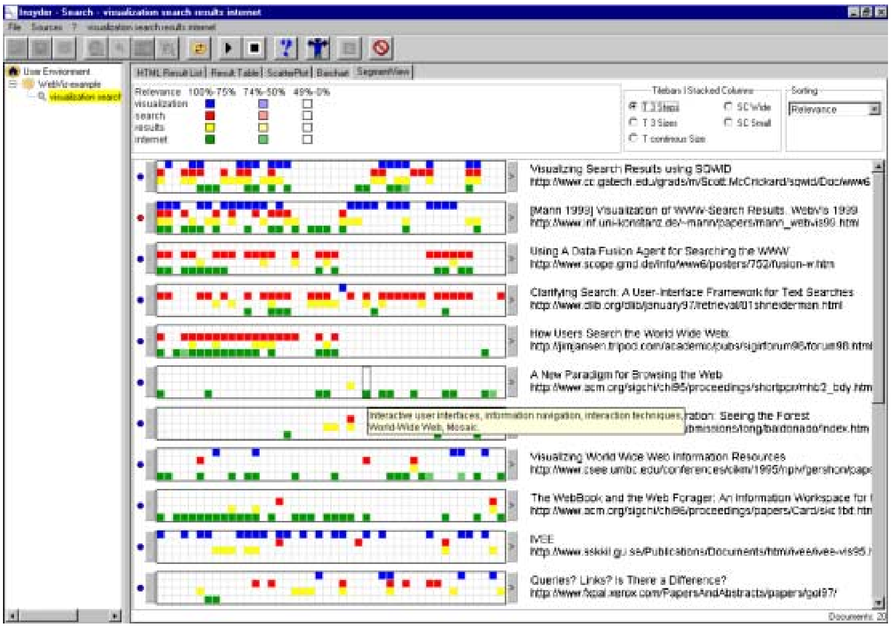
Nb crawled URLS : 0 / 0

Nb crawled search engines URLS : 0 / 0

Fig. 7. Bar graph

of documents, and relevance/server type. The user also has the possibility of selecting his own combination of X and Y dimensions from a subset of variables for each document. The scatterplot thus offers an easy way of navigating through the document space on the set level to find interesting search results. The guidance by predefined combinations also goes in the direction of the attention management rule from [5].

The use of the bar graph (Fig. 7) was inspired by the work of Veerasamy and Belkin [55, 56]. The original idea of bar graphs, showing overall and single keyword relevance using the length of bars, has been adapted in several ways [35]. The colours used for the different keywords are the same as for tile bars and stacked columns. Each row of bars represents one document and shows the distribution of the relevance for each keyword of the query and the total relevance for the document. It is therefore easy to detect if a document deals with one or more of the different keywords of the query. The headings of each column (select flag, relevance, etc.) can be sorted in in-



● T 3 Steps ○ SC Wide
○ T 3 Sizes ○ SC Small
○ T continous Size

Sorting: Relevance

Visualizing Search Results using SQWID
http://www.cc.gatech.edu/grads/m/Scott.McCrickard/sqwid/Doc/www6

[Mann 1999] Visualization of WWW-Search Results. WebVis 1999
http://www.inf.uni-konstanz.de/~mann/papers/mann_webvis99.html

Using A Data Fusion Agent for Searching the WWW
http://www.scope.gmd.de/info/www6/posters/702/fusion-w.htm

Clarifying Search: A User-Interface Framework for Text Searches
http://www.dlib.org/dlib/january97/retrieval/01shneiderman.html

How Users Search the World Wide Web
http://jimjansen.tripod.com/academic/pubs/sigirforum98/forum98.html

A New Paradigm for Browsing the Web
http://www.acm.org/sigchi/chi95/proceedings/shortppr/mhb2_bdy.htm

Interactive user interfaces, information navigation, interaction techniques, World Wide Web, Mosaic

ration: Seeing the Forest
bmissions/long/bardonado/index.htm

Visualizing World Wide Web Information Resources
http://www.csee.umbc.edu/conferences/cikm/1995/npiv/gershon/pape

The WebBook and the Web Forager: An Information Workspace for
http://www.acm.org/sigchi/chi96/proceedings/papers/Card/skc1txt.htm

IVEE
http://www.ssknll.gu.se/Publications/Documents/html/ivee/ivee-vis95.

Queries? Links? Is There a Difference?
http://www.fxpal.xerox.com/PapersAndAbstracts/papers/gol97/

Documents: 20

Fig. 8. Tile bar view with tooltip

creasing or decreasing order. This function offers the user the possibility of viewing the distribution of the relevance of each keyword individually.

▨ Whereas the above-described visualisations aim to give an overview of the document set, the segment view with tile bar (Fig. 8) and stacked column focuses on single documents. The integration of tile bar and stacked column into one component follows the rule of parsimony [5]. For ranking purposes documents are broken down into segments.

The use of tile bars was mainly inspired by the work of [19]. In contrast to the original tile bars, we did not use grey levels to show the keyword relevance for a segment. Instead, each concept is represented with a different colour (the same colour map as used for the bar graph and the stacked columns). Each document is represented by a rectangular bar, which is displayed next to the title of each document. The length of the rectangle indicates the length of the document. The bar is subdivided into rows that correspond to the keywords (concepts) of the query. The bar is also subdivided into columns, each column referring to a segment within the document. Concepts that overlap within the same segment are more likely to indicate a relevant document than concepts that are widely dispersed throughout the document. The darkness of colour (display variant called 3 steps) or the size of the coloured area of each square (display variants called 3 sizes or continuous size) correspond to the relevance of the concept for that segment of text: the darker the colour of the square (tile) or the larger the coloured area of the square, the higher the relevance. A white tile indicates no relevance for the concept. Visual patterns are meant to indicate whether concepts occur as a main topic throughout the document, as a subtopic, or are just mentioned in passing.

4 Information retrieval aspects

One way to enhance the usability was to create a content-based system, one part of which is a KB. Based on its characteristics, it is best compared to an extensive thesaurus, modelled as a semantic net. The KB is multilingual (English, French, Spanish, German) and is created on a manual base, though a number of interfaces to common electronically available thesauri (e.g. MEDLINE) exist. The KB models the real world by a controlled vocabulary, which can be individually adapted to various application domains to meet business demands. The KB consists of concepts (nodes) describing the semantics of the system by using relationships, such as *hyperonyme* (is-a), *associated idea*, *instance of*, *defined by*, *contains implicitly*, *feature of*, *consists of*, *belongs to*, and *translation*. Each concept is in a primary hierarchy relation, defined by *in, out* and the *type* of relationship, which can be rooted to a '*thing*'. In addition to the primary hierarchy, a secondary one exists that describes the wider context of a term (e.g. Internet is also associated with Yahoo!, Hyperspace, HTTP, and so forth). Besides these concepts the KB also contains grammatical forms like nouns, verbs, adjectives, or adverbs and the corresponding lemma form. Like a thesaurus the KB 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 about how useful a knowledge-based retrieval might be in the context of the Web, because a well-known body of knowledge that can be associated with the documents in the Web does not exist. However, findings from query expansion studies showing that a domain-specific knowledge base is beneficial encourage such a use [6].

4.1 Ranking

Two ranking algorithms have been proposed and implemented for the INSYDER system. Both follow an on-the-fly ranking paradigm: Each document is ranked individually and not in relation to a given collection. The standard ranking is called natural language, the second one concept query.

4.1.1 Natural language ranking

The natural language algorithm is the default ranking algorithm of the system. Here, the user types in a query much like 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 by eliminating stopwords and punctuation. This leads to a query consisting of the following terms: 'inhabitants London'. The two meaningful terms extracted from the original query are used for crawling the Web and sent to the different sources, e.g. search engines, chosen by the user, to retrieve a first result. From these results the crawling agents extract the links, which then serve as the basis for further crawling. The results from the crawling agents are given to the ranking component, which calculates a number, describing how well a document found by the crawling agent matches the query. Numbers range from 0 to 100, where 0 means no match at all and 100 means best match possible. Documents above a ranking value of 75 match the query very good, at least in one part of the document in which all query terms occur.

These ranking values are calculated using a semantic analysis algorithm in the background. Documents found are represented in a metadescription, consisting of information about segments (which are usually similar to the sentences) and their description in the KB. The query itself is also put in such a metadescription. This way, the system has a basis for the comparison of the query and the document. The matching is done segment by segment; thus a value for each segment is achieved. This set of values is called the detailed rank (dR). The values of this dR are used for the visualisations. The final ranking value is then calculated, taking into account the mean value of all segments and the maximum value reached in one or more segments. 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 loose order and spread all over the document, the maximum value gets a higher weight in the final calculation than the mean.

$$\begin{aligned} globalRank = & \frac{n * \sum dR[segment]}{numberOfSegment} \\ & + (100 - n) * \max(dR[segment]) \qquad (1) \end{aligned}$$

Natural language ranking (here $n = 25$)

This way, calculating the overall relevance of a document can be compared to a Boolean '*AND*' with a '*NEAR*' proximity operator.

4.1.2 Concept query ranking

The concept query was designed with the aim of giving the user a higher level of influence and interaction, defining his information need. The concept query is not intended for beginners in searching; it is for advanced users. The idea behind this 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. By 'query' we mean that the terms used are *concepts* defined in the KB. The user can select such a concept from the KB by using the visual query not in the sense of an interactive query expansion as with the natural language ranking but to gain access to the concepts stored in the KB. The principle design of the concept query algorithm is the same as that of the natural language algorithm. The difference is the use of concepts from the KB and the user's option to weight these. (See [37] for a detailed description.)

4.2 Classification in INSYDER

Meta-information, as for example demanded by [44], can play a vital role when analysing documents. The aim of the classification methods developed in the current work is to obtain meta-information about the document. This gives the user additional information about the result set and the documents found. Depending on the kind of visualisation, numerous attributes of a document can be visualised. Two types of classifications have been developed, a *content-based* and a *formal* one.

4.2.1 Content-based classification

Unlike sequential text documents, hypertext documents are characterised by contextual units that are linked to each other. Basically one can distinguish two types of Web hypertext documents, those that serve as a *webliography*, meaning that the documents, like a bibliography, contain links to publications, and those that are built upon links to other Web pages and other documents consisting of text. With the increasing spread of interactive tools for the Web, e.g. animated pages, these could be seen as a third category. However, during the development of the content-based classification, these tools played a minor role and were therefore not taken into account.

For the present development, a webliography seemed to be too broad and has therefore been divided into two subclasses: *bookmark list* and *catalogue*. A bookmark list is defined as a webliography where the links are mostly 'outer' links, meaning they point to a different server. By contrast, the catalogue is defined as a webliography where most of the links point to links deeper in the hierarchy of the Web site, i.e. 'inner' links. Besides the classification as a bookmark list and catalogue, a *mixed link list* and *text/images* are foreseen. A mixed link list is a hybrid of a webliography and text/images. A typical example is a document containing many inner and outer links within a descriptive text; here the focus is on the hyperlink aspect. Text/images are defined as anything not falling in the previous categories. For the classification, the relation of inner and outer links and amount of text are analysed and compared to the findings of studies describing a standard Web page [8, 14]. The algorithm in the background is rather simple; based on the number of inner and outer links in a document, it is put in the various categories.

4.2.2 Formal classification

The formal classification takes a formal criterion, the URL of a document, to determine its source, referred to as the server type of the document. The server type is especially useful for business decision makers as they can see at a glance what kind of source the document is coming from, e.g. using a server type definition containing competitors, the user can determine at once if the document is from a competitor, without knowing the URL of the competitor or its name.

The resulting classification allows the user to structure the result set according to the categories identified. Also, using the other visualisations, in particular the scatterplot, becomes extremely helpful when analysing a result set. This way the user can see right away (depending on the server type definition) whether the highest ranked documents are documents from competitors, governmental reports, etc.

4.3 Relevance feedback

Relevance feedback is a commonly accepted method of improving the retrieval effectiveness [9, 27]. User relevance feedback can be seen as an interactive process, where users are encouraged to use their domain knowledge to generate more comprehensive queries [4].

INSYDER's retrieval functions are based very much on the KB. This KB is used for the suggestion of terms for a new 'relevance feedback query'. As INSYDER is not based on a vector space or probabilistic model [48], a new way of deriving feature terms for the relevance feedback had to be found. With the whole system based on the KB, the KB consequently was also used for the relevance feedback. Especially with regard to a domain-specific information need, its adaptability to certain domains promises a more distinct query formulation than the initial query. Therefore, the feature terms describing the documents are defined as the concepts from the KB. As a result of their experience with several user tests on relevance feedback, Koenemann and Belkin [27] propose that the query created be made transparent, so that the user is able to revise queries in an iterative process. Therefore, it has been decided to give the user suggestions for terms for the relevance feedback. The user interface for the relevance feedback is the visual query. Instead of being used for the initial query step, the visual query is involved after the user has deemed certain documents as relevant or irrelevant and the system has proposed a new query based on these judgements. Hence it is an interactive relevance feedback, meaning it uses judgements made by the user about documents (like, dislike) to derive from these new query terms, which are suggested to the user, who can then decide to eliminate some of the concepts proposed. The proposed and implemented relevance feedback for the INSYDER system can be formally described as follows: A document is defined by a number of feature terms $\mathrm{doc}_i = (f_{i1}, f_{i2}, \ldots, f_{in})$. The user judges the documents as relevant or irrelevant:

$$ft_j = \sum_{relevant} f_i - \sum_{irrelevant} f_i \quad \text{with } j = [1 \ldots 20] \tag{2}$$

Proposed relevance feedback with concepts in INSYDER

5 Evaluation of INSYDER

We claim that the INSYDER system enhances usability. As shown above, the usability aspect is manifold. Within the course of the project we evaluated the performance of the retrieval (i.e. its ranking component) and the usability of the visualisation of search results separately.

5.1 Evaluation of the ranking component

Yuwono and Lee [58] argue that *'it is infeasible to search Web pages directly to compute the relevance scores without the help of the index'*. Like the INSYDER system, *Inquirus* [30] also performs an online analysis and its own, and therefore consistent, ranking of documents found by search engines, but it is designed to be a mere metasearch engine.

The objective of the present evaluation was to find out whether the discussed 'on-the-fly' ranking algorithms could deliver valid results. Therefore, two independent tests were conducted. The first test was to use 25 TREC (Text Retrieval Conference) topics for an evaluation of the ranking of Web pages, available in August 2000 (online evaluation). The second test was also to use 25 TREC topics but to use also the test collections with the judged documents provided by TREC (offline evaluation). On both occasions an adapted *idf*-based ranking scheme was used as a baseline for the comparison.

It is well known that other more sophisticated ranking algorithms exist that give a good retrieval performance and could be superior to the current baseline; however, the implementation of these would have exceeded the current evaluation whose aim was to measure the retrieval effectiveness of the 'on-the-fly' ranking.

5.1.1 Evaluation setting

For the online evaluation INSYDER was used as a standalone search engine; the possibility that other sources besides search engines might be used as a first step in crawling was not considered. Therefore, we decided to use the search engines AltaVista, Hotbot, Yahoo!, and Northernlight as starting points. A query was sent to the systems, their outcomes were retrieved and analysed (i.e. a consistent ranking was done) and used for further crawling. From the 50 TREC topics given in TREC-8 (ad hoc and small Web topics), 25 were chosen for the evaluation. In this way the topics chosen were those that the judges were familiar with and that gave variety to the number of search terms per topic (number of query terms-frequency: 1-1;2-13;3-10;4-1). The query terms were in most cases taken from the title of the topic description; in some cases extra query terms were added or altered when it seemed that the query would not lead to relevant documents (derived by the description and the narrative). The assessors were two members of the Information Systems working group of the Department of Information and Computer Science at the University of Konstanz. Documents where one judge was unsure about the relevancy were discussed by both judges. Overall, 5900 documents were downloaded for evaluation. Of these, 500 were analysed to measure the precision of the system. An attempt was made to consider the recommendations of Buckley and Voorhees [10]; however, due to resource restrictions, we limited ourselves to 25 topics.

The effectiveness of IR systems is commonly described by using two measurements: recall and precision. Recall is extremely difficult to measure, as most of the time the total number of relevant documents in a collection is unknown. This is especially so with regard to the Web.

The second (offline) evaluation was conducted using the TREC collection as the result set, with the aim of supporting the findings of the online evaluation and also obtaining insight into the recall performance of the system.

However, the drawback of not being able to use a domain-specific KB applies here.

Based on the research of Sparck-Jones [52], Buckley and Voorhees [10] propose a fuzziness value that resembles the differences in scores between two runs. According to Sparck-Jones [52], a difference of 5% is noticeable. In their analysis of TREC data, Buckley and Voorhees [10] show that the error rate decreases as the fuzziness value increases. However, they also state that fewer conclusions can be drawn from these results, as different (ranking) methods are considered to be equal [10]. As a further conclusion they recommend using average precision as a reasonable measurement, and for environments like the Web to use cut-off levels of 10 or 20. Taking these recommendations as a rule of thumb, for the results of the present evaluation the following setting has been defined:

- Calculation of the precision (Prec) and average precision (AvPrec) at cut-off levels of 20: Prec(20), AvPrec(20)
- A fuzziness value of 5%
- An error rate as the quotient of the minimum number of errors (i.e. the times one system performed better than the other or vice versa) and the total number of decisions (here the number of topics)

5.1.2 Results

Using the premises as described earlier and the AvPrec(20) measurement it was found that the natural language algorithm (INSYDER ranking) performed better than the tfidf-based ranking value 11 times, both performed equally well 3 times, and the natural language algorithm performed worse 11 times. A similar result was obtained using the Prec(20) measurement. Here the natural language algorithm performed better 9 times, both performed equally well 10 times, and in 6 cases the natural language algorithm performed worse.
The analysis of the result of the offline evaluation (using the TREC collection) followed the same methodology as the online evaluation. As the number of relevant documents was known this time, the recall measurements could also be given. As with the former evaluation a cut-off level of 20 and a fuzziness value of 5% were set. Figure 9 shows the precision-recall graphs for the two methods at different cut-off levels. The graphs show that after having viewed 25 documents a third of all relevant documents have been seen.

The premise for the concept query was to obtain more precise results as the query specification was already very precise. A comparison of the concept-query-based ranking with the natural-language-based ranking seems to verify this, taking four topics as a basis. However, the concept query performs much worse than the natural language ranking when it comes to recall.

The retrieval showed that the on-the-fly ranking as performed by the INSYDER system performs well in terms of its retrieval effectiveness. Generally, it achieves better results than the simple tfidf-based ranking scheme. The natural language ranking is superior to the concept query in terms of general use, although the assumption that the concept query leads to more precise results could

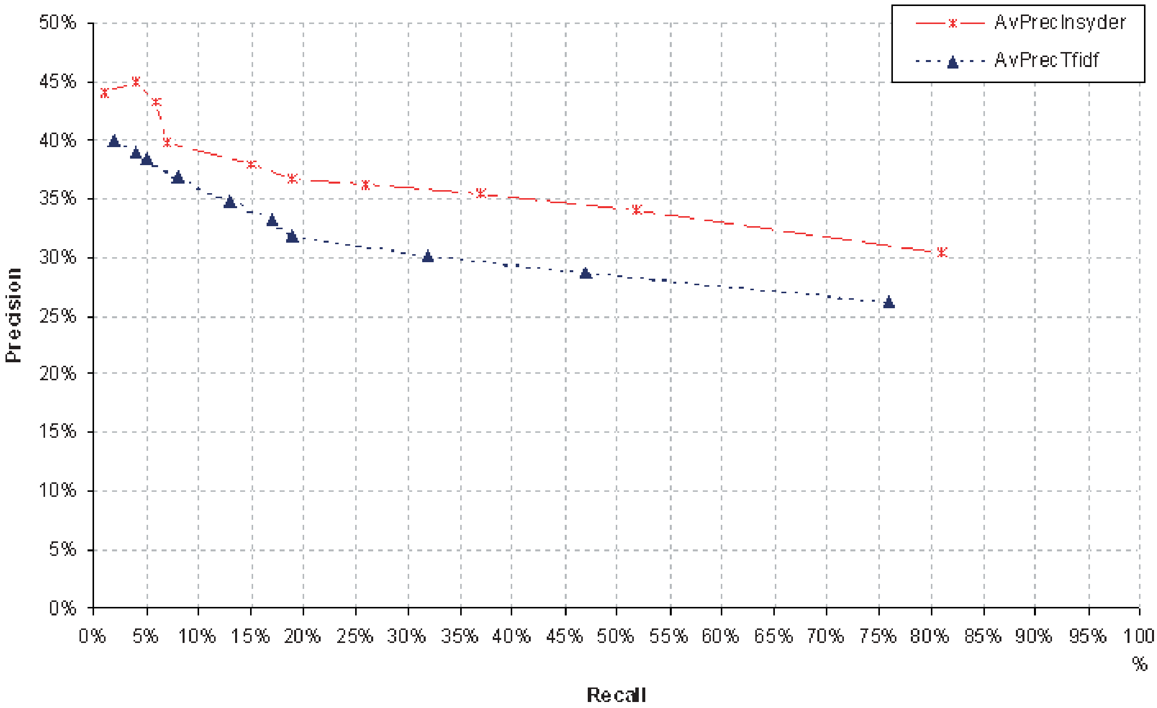


Fig. 9. P-R graph for the comparison of INSYDER and tfidf-based ranking, based on average values at cut-off levels: 1, 2, 3, 5, 10, 15, 20, 30, 50, 100

be emphasised in a small evaluation. The current test setting using the TREC data was chosen to have a stable and reliable test environment; however, it does not reflect the daily business use of the INSYDER system as the topics chosen are not the particular information needs of business decision makers. For such a test setting an adapted KB and 'real' information needs from the user group would have been required. However, results would not be comparable as the current evaluations are using the established setting provided by TREC.

5.2 Evaluation of the visualisations

The evaluation was conducted as a user test with 40 users and focused on the different visualisations used to present the search results in the result phase of the search process. The usability evaluation part of the study was focused on the added value of the visualisations in terms of their effectiveness (accuracy and completeness with which users achieved task goals), efficiency (the task time users expended to achieve task goals), and subjective satisfaction (positive attitudes to the use of the visualisation) for reviewing Web search results. Assuming the advantages of a multiple-view approach described in the literature, we did not intend to measure the effects of using scatterplots, bar graphs, and segment views instead of the list and table. We wanted to see the added value of using these visualisations in addition to the result table. Another goal of this evaluation was to measure the influence of the factors' target user group, type and number of data, and task to be done on the effectiveness, efficiency, and user satisfaction for each visualisation. Table 1 gives an overview of the main hypothesis behind the evaluation. In what follows we will focus only on the first hypothesis. For a detailed explanation refer to [35] and [32].

Table 1. Main hypothesis of the INSYDER visualization evaluation

Hypothesis
The result table and the visualisations produce results in terms of usability that differ from the results for the HTML list.
The target user group influences how the usability will be determined by the user interface condition in comparison with the HTML list.
The task type influences how the effectiveness will be determined by the user interface condition in comparison with the HTML list.
The number of documents presented influences how the effectiveness will be determined by the user interface condition in comparison with the HTML list.
The number of query keywords used and shown influences how the effectiveness will be determined by the user interface condition in comparison with the HTML list.

5.2.1 Independent variables

5.2.1.1 User interface. From the various possible settings and combinations of components, the following five user interface configurations were tested:

– HTML list only
– Result table only
– Scatterplot + Result table
– Bar graph + Result table
– Segment view + Result table

Today search results are traditionally presented in the form of a list. Therefore, the HTML list was considered as the standard against which the visualisations would be compared. It was included in the evaluation as a baseline for the usability values.

5.2.1.2 Target user group. As described above, the target user group for the INSYDER system were business analysts from small- and medium-sized enterprises. Nonetheless, the summative evaluation of the visualisations was done with students from different disciplines and university staff. This decision was possible because searching the Web is an activity not restricted to the special target user group of the INSYDER system. Moreover, most of the special functions of the INSYDER system created for use in the context of business intelligence played only a marginal role in this evaluation. Last but not least, business graphics are quite common in everyday life, and the visualisations implemented are simply compared to many other ideas found in the literature.

5.2.1.3 Task. In order to observe possible influences caused by the task to be done, we decided to use two of the four different types of information-seeking tasks described in [50]. Half of the tasks that the users had to fulfil were of the type 'specific fact finding'; the other half were of the type 'extended fact finding'. A typical task for specific fact finding was for example 'How long is the Danube river', searched with the keyword 'danube' and a result set size of 30. An extended fact-finding task was for example 'List mountains that are higher than 8000 meters (26 248 feet)!' with keywords 'mountain himalaya altitude height top peak reinhold messner' and a result set size of 500.

The general concept behind the evaluation was to concentrate on the information-seeking process of the phase, referred to variously as review of results, evaluate results, or examine results. Therefore, we created an information need for the user by asking a question. The user then had to skip several steps because we had already performed them for all users so as to eliminate influences from these phases. In the review of results, we constrained the user by not allowing steps like reformulation of the query and selection of other sources, and we forbade browsing.

5.2.2 Dependent variables

To measure the effectiveness and the efficiency of the visualisations, we used high-level metrics as defined by [12].

Findings from other authors evaluating visualisations such as Sen and Boe [49] strengthened our motivation to measure, in addition to the performance facts (effectiveness, task time, and temporal efficiency), the soft facts (expected added value and satisfaction).

5.2.2.1 Effectiveness. In general, effectiveness can be measured by the accuracy and the completeness with which users achieve the goals of the test tasks [22]. In the case of our evaluation, the effectiveness was measured as a completion rate scored on a scale of 0 to 100%.

5.2.2.2 Task time. The task time was measured in seconds from the moment the result set was opened until the question was successfully answered, the user terminated the task, or the time limit to complete each test task was reached. The time taken to read and understand the task itself was not measured. Drawing on the results of a pretest, the time to answer specific fact-finding questions was limited to 5 min per question, for extended fact-finding tasks to 10 min per question.

5.2.2.3 Temporal efficiency. The temporal efficiency was calculated as effectiveness divided by task time.

5.2.2.4 Expected added value. For all conditions where the users were free to utilise the result table and/or one of the three visualisation components (scatterplot, bar graph, or segment view), we sought to measure how these multiple view choices would be used. We therefore measured the usage times separately for each component and calculated the proportion of usage time between the visualisation and the result table in the visualisation plus result table conditions. In addition, we calculated which tasks had been solved by using the result table alone, the visualisation plus the result table, or the visualisation alone. Our thinking was that the users would utilise the visualisation to support their task based on their expectations about the added value.

5.2.2.4 Satisfaction. We measured satisfaction with a questionnaire. Test users were asked to rate their satisfaction in terms of: ease of use, self-descriptiveness, suitability for learning, layout, suitability for the tasks, and conformity with expectations.

5.2.3 Procedure

The overall test procedure for every subject included the five main steps and lasted approximately 2 h:

- Entry questionnaire
- System presentation
- Warm-up (free exploration of the system)
- Test tasks
- Closing questionnaire

5.2.4 Test tasks

Each of the 40 users had to perform 12 tasks. All users processed the same 12 questions with the same keywords and number of hits in the same order. The difference between the five groups was the visualisation that the user could use to answer the question. The system ensured that for each task that a user had to fulfil he could only see the result set and visualisations provided for this step. The setting for this controlled experiment assured that the five combinations of visualisations were distributed equally among all variables.

During the tasks, the users were requested to ‘think aloud’ so as to enable the evaluation team to understand and record their current actions.

5.2.5 Results

5.2.5.1 Expected added value. As regards the user interface the evaluation setting contained two conditions where the users were forced to use a single component: the HTML list or the result table. In three conditions, the users were free to use the result table only, the visualisation plus result table, or just the visualisation. Despite the fact that visualisations were available, the users decided in about 28% of the tasks to use only the result table: 25% in the scatterplot condition, 36% in the bar graph condition, and 24% in the segment view condition. When users had the choice between the scatterplot and the result table, they spent 34% of the time using the scatterplot and 66% the result table. The values for the bar graph were 30%/70%, for the segment view 44%/56%. The users always spent less time for the visualisation than for the result table. The result table was the favourite component of the users, as the analysis of the usage times shows. It was used in all three user interface conditions with scatterplot, bar graph, and segment view more than 50% of the overall task time. Interpreting usage time as an indicator, the expected added value for the users of the result table seemed to be higher than that of the other components. Usage time of a component could be a misleading indicator for expected added value, because it is possible that usage of the component is necessary for a certain task, despite its not being favoured by the user. When combined with the results from the questionnaire, usage time may be an indicator of expected added value. According to usage time ratios the result table has the highest expected added value, followed by the segment view, the bar graph, and the scatterplot.

5.2.5.2 Summary of the questionnaire results. The favourite component of the users was the result table, followed by the segment view and the HTML list. The bar graph and especially the scatterplot found little favour. The ranking of the components derived from the questionnaire was the same as that derived from the usage times. The users' order of preference was the result table followed by the segment view, the bar graph, and the scatterplot.

One of the hypotheses examined on the basis of the questionnaire results was that the result table and the visualisations produce results in terms of user satisfaction that differ from the results for the HTML list. Compared with the HTML list, the result table and the segment view received higher mean user satisfaction values, the scatterplot and the bar graph lower. The differences in the user satisfaction indexes between the HTML list and the values from the other visualisations were calculated and tested using a t-test for dependent samples. Table 2 shows the t-values and the corresponding probability values of the two-tailed paired t-test.

The user satisfaction for the result table is significantly higher than that for the HTML list on the 1% significance level. In the same direction but not significant is the difference between the segment view and the HTML list ($p < 0.1$). The values for the scatterplot and the bar graph, which performed worse than the HMTL list, are far from significant. The hypothesis was therefore confirmed only for the result table. The result table produced results in terms of user satisfaction that differ from the results for the HTML list.

5.2.5.3 Summary of the performance facts results. When summarising the performance facts results the temporal efficiency of the different components in the different situations may be a good indicator [32].

In general, the result table and the visualisation plus result table conditions all showed more or less lower mean effectiveness index values and at the same time higher mean task time index values than the HTML list (Table 3). The lower effectiveness in comparison with the HTML list is only significant for the bar graph condition. The differences for the effectiveness values of the other user interface conditions are not significant.

The task time index differences in comparison with the HTML list were significant for all user interface conditions (Table 4). For the scatterplot condition, the effect was highly significant. None of the user interface conditions showed higher mean effectiveness values than the HTML list, and all had significantly higher mean task time values. The temporal efficiency of the result table and the visualisation plus result table conditions may therefore not be better than that of the HTML list. A detailed analysis of the temporal efficiency for three selected questions revealed that in all except one case, the result table and the visualisation plus result table conditions showed a lower mean temporal efficiency than the HTML list. The exception was that in one of the three questions, the bar graph condition performed better than the HTML list. A statistical validation showed that most of the differences found were not statistically significant at this level of analysis. Significant differences were only found for the result table in one question. Highly significant differences were found for the scatterplot condition in two of the three questions. Despite the missing statistical significance at this detailed level of analysis, the results for these tasks support the overall impression that the temporal efficiency of the tested visualisations of search results is worse than that of the traditional HTML list.

Table 2. Two-tailed paired t-test: user satisfaction index

Reference	Compared visualisation	t-value	Pr > |t|
HTML list	Result table	-4.42	$< 0.0001^{**}$
HTML list	Scatterplot	1.19	0.2430
HTML list	Bar graph	0.46	0.6471
HTML list	Segment view	-1.82	0.0765

** Highly significant ($p < 0.01$)

The results of the evaluation have been an important step on the way to more insights into the usability of visualisations of search results from the World Wide Web. Based on these insights we have totally redesigned the INSYDER system. The main redesign ideas will be presented in the following section.

6 Outlook

For possible improvements of the components and the overall INSYDER system, two ideas in particular will be discussed:

- The integration of result table, HTML list, bar graph, segment view, and, as an additional element, a table lens into one component called supertable.
- Improvements to the scatterplot component.

Table 3. Two-tailed paired t-test: effectiveness indexes for the user interface conditions

Reference condition	Tested condition	t-value	Pr > |t|
HTML list	Result table	0.15	0.8824
HTML list	Scatterplot + Result table	0.80	0.4307
HTML list	Bar graph + Result table	2.05	0.0471^{*}
HTML list	Segment view + Result table	1.27	0.2128

* significant ($p < 0.05$)

Table 4. Two-tailed paired t-test: task time indexes for the user interface conditions

Reference condition	Tested condition	t-value	Pr > |t|
HTML list	Result table	−2.28	0.0285*
HTML list	Scatterplot + Result table	−4.11	0.0002**
HTML list	Bar graph + Result table	−2.49	0.0171*
HTML list	Segment view + Result table	−2.16	0.0373*

* Significant ($p < 0.05$)
** highly significant ($p < 0.01$)

The users requested a number of features for the bar graph and the segment view already implemented in the result table. On the other hand they requested that the document tooltip from these two components also be available in the result table. Even with the existing result table the user satisfaction values for the result table were significantly better than for the HTML list. The result table performed nearly as good as the HTML list in terms of effectiveness and task time. Differences may have been caused by the fact that the document extract was not fully visible and that line breaks enhancing the readability of the text had not been used in contrast to the HTML list. In addition, the HTML list had common HTML-navigation elements. All this could also be implemented in the result table.

There are already components existing that integrate smoothly a result table and a bar graph, e.g. the *Table Lens* [43]. Additionally, the result table of the INSYDER system already contains a simple predecessor of the segment view in the form of the relevance curve. So the proposed supertable would integrate a multiple-focus table lens, the HTML list, a bar graph and the segment view with tile bars and stacked column in a way that allows easy manipulation of the table. This could be done, for example, by including a set of buttons like those already implemented for the scatterplot. Instead of predefined combinations of dimensions and axes, there would be different predefined zoom levels. The first zoom level will be a bar graph view, showing only numerical data (e.g. relevancies) as bars. The second zoom level will show revealing document details in a one-line-per-document mode for the whole table (e.g. numbers for the relevancies, title, URL). The third zoom level will show in a several-line-per-document mode more details for the whole table (e.g. abstract of the document). In the fourth level the bar graph will be changed by a more detailed tile bar, showing for each document segment details (e.g. text of the whole segment). The user can now browse through the whole document segment by segment, guided by the visual relevance information shown in the tile bar. In addition, tooltips or lens mechanisms could allow easy inspection of details for single documents. Power users could use multiple-focus possibilities for comparisons. The SuperTable as an integrated component would be more complex than the former single components. With adequate design a subject who is starting to use the system might not be aware of this complexity, and therefore would not be distracted. The integration would minimise context switching and would be able to allow a smooth learning curve from beginner mode to power user mode. The evaluation of such a highly integrated component might be demanding, but expectations of possible improvements concerning effectiveness, efficiency, and user satisfaction would be high.

Of the five visualisation ideas implemented in the INSYDER system, four were integrated into the supertable. Because of its two-dimensional layout the scatterplot was left out. Nevertheless, improvements are also possible. Some examples include packing more information into the display, for example, by using partially filled circles in the scatterplot, as done by [16]. If this is done, the overall relevance of a document will always be visible, even when the dimension is not mapped to one of the axes. With the callout lens from the same authors, an exploder for document groups could be integrated allowing fulfillment of one of the requests formulated by the user. With this and other enhancements the scatterplot could serve its functions by giving the user a quick overview of all search results, offering a good variety of controls (e.g. defining own views, zooming, selecting) to reduce the amount of hits to a smaller group of interesting documents. These could then be selected and analysed in more detail in the supertable.

In the meantime an enhanced scatterplot with additional lens mechanisms tightly coupled with the described supertable and a document browser have been implemented. We called this successor of INSYDER *VisMeB* (Visual Metadata Browser). For more information see [26].

References

1. Ahlberg C, Shneiderman B (1994) Visual information seeking: tight coupling of dynamic query filters with starfield displays. In: Adelson B, Dumais S, Olson, JS (eds) Proceedings of the conference on human factors in computing systems, Boston, 24–28 April 1994. ACM Press, New York, pp 313–317
2. Ahlberg C, Wistrand E (1995) IVEE: An information visualization and exploration environment. In: Gershon N, Eick S (eds) Proceedings of the IEEE conference on information visualization, Atlanta, GA, 30–31 October 1995. IEEE Press, Los Alamitos, CA, pp 66–73
3. ARISEM. Retrieved 2003-09-10. `http://www.arisem.com`

4. Back J, Oppenheim C (2001) A model of cognitive load for IR: implications for user relevance feedback interaction. Inf Res 6(2). http://InformationR.net/ir/6-2/ws2.html [retrieved 17 Sept 2003]
5. Baldonado M, Woodruff A, Kuchinsky A (2000) Guidelines for using multiple views in information visualization. In: Di Gesù V, Levialdi S, Tarantino L (eds) International workshop on advanced visual interfaces (AVI 2000), Palermo, Italy, 23–26 May 2000. ACM Press, New York, pp 110–119
6. Beaulieu M (1997) Experiments on interfaces to support query expansion. J Document 53(1):8–19
7. Belkin N, Oddy R, Brooks H (1982) ASK for information retrieval: I. J Document 38(2):61–71
8. Bray T (1996) Measuring the Web. http://dx.doi.org/10.1016/0169-7552(96)00061-X [retrieved 17 Sept 2003]
9. Buckley C, Salton G, Allan J (1994) The effect of adding relevance information in a relevance feedback environment. In: Croft B (ed) Proceedings of the 17th annual international ACM SIGIR conference on research and development in information (Retrieval SIGIR '94), Dublin, UK, 3–6 July 1994. Springer, Berlin Heidelberg New York, pp 292–300
10. Buckley C, Voorhees E (2000) Evaluating evaluation measure stability. In: Belkin N, Ingwersen P, Leong M (eds) Proceedings of the 23rd annual international ACM SIGIR conference, Athens, Greece, July 2000. ACM Press, New York, pp 33–40
11. Choo C, Detlor B, Turnbull D (1999) Information seeking on the Web – an integrated model of browsing and searching. – preprint – In: Proceedings of the 62nd annual ASIS meeting, vol 36, Washington, DC, 31 October–4 November. Information Today, Medford, NY (preprint). http://choo.fis.utoronto.ca/fis/respub/asis99/default.html [retrieved 23 July 1999]
12. Cugini J (2000) Presenting search results: design, visualization and evaluation. In: Workshop on information doors – where information search and hypertext link (in conjunction with ACM Hypertext and Digital Libraries conferences), San Antonio, TX, 30 May 2000. http://www.ics.mq.edu.au/∼einat/info_doors/program/Cugini.html [retrieved 7 July 2000]
13. Eibl M (2000) Visualisierung im Document Retrieval: theoretische und praktische Zusammenführung von Softwareergonomie und Graphik Design. Forschungsberichte/IZ, Informationszentrum Sozialwissenschaften, GESIS Bd. 3. Bonn
14. Fagrell H, Sørensen C (1997) It's Life Jim, But Not As We Know It! In: IRIS'20, Social Informatics, Norway, 9–12 August 1997. http://iris.informatik.gu.se/conference/iris20/35.htm [retrieved 14 Sept 2003]
15. Fairchild K, Poltrock S, Furnas G (1988) SemNet: three-dimensional representations of large knowledge bases. In: Guindon R (ed) Cognitive science and its applications for human-computer interaction. Erlbaum, Hillsdale, NJ, pp 201–233
16. Fishkin K, Stone M (1995) Enhanced dynamic queries via movable filters. In: Katz IR, Mack RL, Marks L et al (eds) Proceedings of the conference on human factors in computing systems (CHI 1995), Denver, 7–11 May 1995. ACM Press, New York, pp 23–29
17. Fowler R, Wilson B, Fowler W (1992) Information navigator: an information system using associative networks for display and retrieval. Technical Report NAG9-551, Department of Computer Science University of Texas – Pan American, Department of Computer Science, #92–1
18. Hasan M, Golovchinsky G, Noik E, Charoenkitkarn N, Chignell M, Mendelzon A, Modjeska D (1995) Browsing local and global information. In: Proceedings of CASCON IBM, Canada Centre for Advanced Studies Conference
19. Hearst M (1995) TileBars: visualization of term distribution information in full text information access. In: Katz IR, Mack RL, Marks L et al (eds) Proceedings of the conference on human factors in computing systems (CHI 1995), Denver, 7–11 May 1995. ACM Press, New York, pp 59–66
20. Hearst M, Karadi C (1997) Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In: Belkin NJ, Narasimhalu AD, Willett P (eds) Proceedings of the 20th annual international ACM SIGIR conference on research and development in information retrieval, Philadelphia, 27–31 July 1997. ACM Press, New York, pp 246–255
21. Hendley R, Drew N, Wood A (1995) Narcissus: visualizing information. In: Gershon N, Eick SG (eds) Proceedings of the IEEE conference on information visualization, Atlanta, 30–31 October 1995. IEEE Press, Los Alamitos, CA, pp 90–96
22. ISO/DIS 9241-11.2 (1997) Ergonomic requirements for office work with visual display terminals (VDTs): Part 11: Guidance on usability
23. Jansen B, Spink A, Saracevic T (2000) Real life, real users, and real needs: a study and analysis of user queries on the Web. Inf Process Manage 36:207–227
24. Jones S (1999) VQuery: a graphical user interface for Boolean query specification and dynamic result preview. http://www.cs.waikato.ac.nz/∼stevej/Research/PAPERS/dl98.pdf [retrieved 9 Dec 2003]
25. Karlgren J, Franzén K (1997) Verbosity and interface design. SICS Technical report T2000, 4 March 1997. http://www.ling.su.se/staff/franzen/irinterface.html [retrieved 12 Sept 2003]
26. Klein P, Müller F, Reiterer H, Limbach T (2003) Metadata visualization with VisMeB. In: Proceedings of the 7th international conference on information visualisation (IV 03). IEEE Press, New York, pp 600–605
27. Koenemann J, Belkin N (1996) A case for interaction: a study of interactive information retrieval behavior and effectiveness. In: Tauber MJ, Bellotti V, Jeffries R et al (eds) Proceedings of the conference on human factors in computing systems (CHI 1996), Vancouver, BC, Canada, 13–18 April 1998. ACM Press, New York, pp 205–212
28. Kohonen T, Kaski S, Lagus K, Salojärvi J, Honkela J, Paatero V, Saarela A (2000) Self organization of a massive document collection. IEEE Trans Neural Netw 11(3): 574–585
29. Korfhage R (1991) To see, or not to see – is that the query? In: Proceedings of the 14th annual international ACM/SIGIR conference on research and development in information retrieval. ACM Press, New York, pp 134–141
30. Lawrence S, Giles L (1998) Context and page analysis for improved Web search. IEEE Internet Comput 2(4):38–46
31. Lokuge I, Ishizaki S (1995) GeoSpace: an interactive visualization system for exploring complex information spaces. In: Proceedings of the conference on human factors in computing systems (CHI 95), pp 409–414. http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/il_bdy.htm [retrieved 10 Sept 2003]
32. Mann TM (2002) Visualization of search results from the World Wide Web. PhD dissertation, Department of Computer and Information Science, Universität Konstanz. urn:nbn:de:bsz:352-opus-7517
33. Mann TM (1999) Visualization of WWW search results. In: Cammelli A, Tjoa AM, Wagner RR (eds) Proceedings of the 10th international workshop on database and expert systems applications, Florence, Italy, 1–3 September 1999. IEEE Press, Los Alamitos, CA, pp 264–268
34. Mann TM, Reiterer H (1999) Case study: a combined visualization approach for WWW search results. In: IEEE symposium on information visualization. Late Breaking Hot Topics Proceedings. Supplement to: Will G, Keim D (eds) Proceedings of the 1999 IEEE symposium on information visualization (InfoVis'99), San Francisco, 24–29 October 1999. IEEE Press, Los Alamitos, CA, pp 59–62
35. Mann TM, Reiterer H (2000) Evaluation of different visualization of WWW search results. In: Tjoa AM, Wagner RR, Al-Zobaidie A (eds) Proceedings of the 11th international workshop on database and expert systems applications, London, 4–8 September 2000. IEEE Press, Los Alamitos, CA, pp 586–590
36. Mayr S (1997) SearchVis: visualising search result sets using a force-based method to form clusters of similar documents. Master's thesis, Graz University of Technology, Austria. ftp://ftp.iicm.edu/pub/theses/smayr.pdf [retrieved 17 Sept 2003]

37. Mußler G (2002) Retrieving business information from the WWW. PhD dissertation, Unversität Konstanz, Department of Computer and Information Science. urn:nbn:de:bsz:352-opus-8447
38. Mußler G, Reiterer H, Mann TM (2000) INSYDER – Information retrieval aspects of a business intelligence system. In: Knorz G, Kuhlen R (eds) Proceedings des 7. Internationalen Symposiums für Informationswissenschaft, Darmstadt, Germany, 8–10 November 2000 8–10 (pp 127–143) Konstanz, Germany: UVK Universitätsverlag Konstanz
39. Nielsen J (1997) Search and you may find. `http://www.useit.com/alertbox/9707b.html` [retrieved 14 Sept 2003]
40. Nikolai R, Kramer R, Steinhaus M (1999) GenThes: a general thesaurus browser for Web-based catalogue systems. A step towards component based catalogue systems. In: 3rd IEEE Meta-Data conference, 6–7 April 1999, Bethesda, MD
41. Nowell L, France R, Hix D (1996) Visualizing search results: some alternatives to query-document similarity. In: Frei HP, Harman DK, Schäuble P et al (eds) Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, Zürich, Switzerland, 18–22 August 1996, ACM Press, New York, pp 67–75
42. Pollock A, Hockley A (1997) What's wrong with Internet searching? In: D-Lib Mag 3(3). `http://www.dlib.org/dlib/march97/bt/03pollock.html` [retrieved 15 March 2003]
43. Rao R, Card SK (1994) The Table Lens. Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: Adelson B, Dumais S, Olson JS (eds) Proceedings of the conference on human factors in computing systems (CHI 1994), Boston, MA, 24–28 April 1994. ACM Press, New York, pp 318–322
44. Rao R, Pedersen J, Hearst M (1995) Rich interaction in the digital library. Commun ACM 38(4):29–39
45. Reiterer H, Mußler G, Mann TM (2000) INSYDER – An information assistant for business intelligence. In: Belkin NJ, Ingwersen P, Leong MK (eds) Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, Athens, Greece, 24–28 July 2000, ACM Press, New York, pp 112–119
46. Reiterer H, Mußler G, Mann TM (2001) A visual information seeking system for Web search. In: Oberquelle H, Oppermann R, Krause J (eds) Mensch & Computer conference, Bad Honnef, Germany, 5–8 March 2001. Teubner, Stuttgart, pp 297–306
47. Robertson G, Mackinlay J, Card S (1991) Cone trees: animated 3D visualizations of hierarchical information. In: Robertson SP, Olson GM, Olson JS (eds) Proceedings of the conference on human factors in computing systems (CHI 1991), New Orleans, 27 April–2 May 1991. ACM Press, New York, pp 189–194
48. Salton G, Buckley C (1997) Improving retrieval performance by relevance feedback. In: Sparck-Jones K, Willett P (eds) Readings in information retrieval. Morgan Kaufmann, San Francisco
49. Sen T, Boe W (1991) Confidence and accuracy in judgements using computer displayed information. Behav Inf Technol 10(1):53–64
50. Shneiderman B (1998) Designing the user interface: strategies for effective human-computer interaction, 3rd edn. Addison-Wesley, Reading, MA
51. Shneiderman B, Byrd D, Croft B (1999) Clarifying search: a user-interface framework for text searches. D-Lib Mag 3(1). `http://www.dlib.org/dlib/january97/retrieval/01shneiderman.html` [retrieved 17 Aug 1999]
52. Sparck-Jones K (1974) Automatic indexing. J Document 30:393–432
53. Spoerri A (1995) InfoCrystal: a visual tool for information retrieval. PhD thesis, MIT, Cambridge, MA. `http://www.scils.rutgers.edu/~aspoerri/InfoCrystal/InfoCrystal.htm` [retrieved 17 Sept 2003]
54. Sutcliffe AG, Ennis M, Watkinson SJ (2000) Empirical studies of end-user information searching. J Am Soc Inf Sci 51(13):1211–1231
55. Veerasamy A (1997) Visualization and user interface techniques for interactive information retrieval systems. PhD thesis, Georgia Institute of Technology, Atlanta
56. Veerasamy A, Belkin N (1996) Evaluation of a tool for visualization of information retrieval results. In: Frei HP, Harman DK, Schäuble P et al (eds) Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, Zürich, Switzerland, 18–22 August 1996. ACM Press, New York, pp 85–92
57. Wood A, Beale R, Drew N (1995) HyperSpace: Web browsing with visualisation. In: WWW3: 3rd international World Wide Web conference, Darmstadt, Germany. `http://www.cs.bham.ac.uk/~amw/hyperspace/www95` [Retrieved 16 Nov 1998]
58. Yuwono B, Lee D (1996) Search and ranking algorithms for locating resources on the World Wide Web. In: Proceedings of the 12th international conference on data engineering, New Orleans, pp 164–171
59. Zizi M, Beaudouin-Lafon M (1994) Accessing hyperdocuments through interactive dynamic maps. In: Proceedings of the 1994 ACM European conference on hypermedia technology, pp 126–134