HAL open science

Between browsing and search, a new model for navigating through large documents

Thibault Mondary, Amanda Bouffier, Adeline Nazarenko

To cite this version:

Thibault Mondary, Amanda Bouffier, Adeline Nazarenko. Between browsing and search, a new model for navigating through large documents. EuroCogSci07, The European Cognitive Science Conference 2007, May 2007, Delphi, Greece. pp.634-639. hal-00153435

HAL Id: hal-00153435

https://hal.science/hal-00153435v1

Submitted on 11 Jun 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Between browsing and search, a new model for navigating through large documents

Thibault Mondary, Amanda Bouffier Adeline Nazarenko
(firstname.name@lipn.univ-paris13.fr)
Computer Science Lab of the Paris 13 University (CNRS UMR 7030)
99, avenue J.-B. Clément, F-93430 Villetaneuse, France

Abstract

This paper proposes a new model for document access, which combines the search and browsing approaches. We define a good navigation as a navigation which is as quick and direct as possible and which offers good precision and recall rates in finding the text segments that are relevant for the user's information need. Our navigation model relies on recent advances in natural language processing and it is based on two traditional cognitive principles that are inherited from works on the visualization of information. This model is implemented in a navigation prototype which is designed for physicians who want to consult official recommendations to take medical decisions. Even if this model has not been really evaluated yet, we show the dynamicity and the efficiency of our approach on few detailled examples.

Introduction

Developing methods and tools that help users to get access to the content of documents is a challenging task as the volume and heterogeneity of accessible information are constantly increasing. In this paper we are considering a specific medical task. Medical organizations such as HAS or AFSSAPS[1] are producing recommendations to help physicians in their diagnosis and treatment tasks. These recommendations take the form of 10 to 50 pages documents. A sample of the diabetes care guideline is given below.

> I. Diabetes Care
>
> I.1. Lifestyle modification
>
> In a first step, patients should receive individual advice on nutrition. Whenever possible, they should be referred to a dietitian who will assess their current intake and nutrition needs.

Figure 1: Extract from of a diabete care guideline

Unfortunately, these recommendations are seldom used: it takes too much time to refer to such a large and complex document to find a precise piece of information dealing with a specific patient case.

[1]Haute Autorité de Santé and Agence Française de Sécurité Sanitaire des Produits de Santé.

In this paper we present a navigation model that has been designed to facilitate the consultation of these medical recommendations by physicians. Specific constraints have been taken into consideration. Navigation *quickness* is important as the physicians must be able to refer to recommendations during their consultations. The navigation tool must guide the user towards all the text segments that are relevant to his/her information need (*relevance* and *exhaustivity*). The tool must be *reliable* as it supports medical decisions: any short paragraph dealing with a specific drug interaction or a rare complication of a given disease may be important.

Traditional information access tools fall into two different categories, which focus either on search or browsing. We argue that recent advances in natural language processing (NLP) now make it possible to combine these two information access means into a unique navigation tool that offers a better compromise between quickness, relevance, exhaustivity and reliability.

The first section presents the various methods that have been proposed to give access to document content. The second and third sections describe our navigation method and the prototype in which it has been implemented. The last section shows the navigation process through the detailed analysis of few medical queries.

Methods for textual information access

Many tools have been designed to facilitate the document access, over the last two decades. Our own approach stems in three different research domains: robust text mining, information visualization and natural language processing.

Text mining

Text mining tools differ along two main axes.

The first one deals with the size of the document base and consequently the granularity of the information needs addressed. Search engines are designed to cope with numerous and usually heterogeneous collections of documents [Baeza-Yates and Ribeiro-Neto, 1999]. General purpose web search engines rely on a very simple representation of the document content and allow any kind of keyword queries. They may also take into account the hypertextual structure of the document collection which is the web. Such engine support only one part of the information access process. The user still has 1) to select some documents out

of the list of returned documents and 2) to read or browse them. On the opposite side, traditional devices like tables of contents or back-of-the book indexes help the reader to find out interesting segments in a single document. Some indexing tools are designed to give such a content-based access to digital documents [Chi et al., 2004, Zargayouna et al., 2006]. Between these extremes, one can find various solutions, ranging from domain specific or intranet search engines to web site or multidocument indexes [Anick, 2001].

The second axis is more important from a cognitive point of view. It deals with the diversity of the users' information needs [O'Hara, 1996]. Search tools help users to find rapidly an answer but they must be able to word their information need in the form of a question or a keyword query. Beyond search engines, question-answering systems aim at giving a precise answer to the user's question (*e.g.* "Roma" for the question "In which city is the Coliseo?") [Burger and al, 2002]. Browsing is less focused. Browsing tools are designed for users that are interested in a document but without any precise information need in mind. They help users to locate interesting segments or to get a general view of what the document is about. Hypertextual links are a typically designed for browsing documents. Abstracts, tables of contents, explicit indexes are alternative means to get an overview of the document content.

Search favors the quickness and relevance of information access but the user has to blindly rely on the system selection. Browsing gives a broader access to information, let the users make their own choices and favours information serendipity, with the risk for the users to get lost (cognitive disorientation [Conklin, 1987]) or to spend too much time wandering around the interesting segments.

Search and browsing should not be opposed, however. Except for very specific information needs which can be easily and unambiguously answered by the document, search must be complemented with browsing. The search engine user usually has to browse the list of returned documents, which is sometimes only used as a starting point for hypertextual navigating through the collection. Most question-answering systems deliver an answer together with the document passage from which it has been extracted so that the user can interpret the answer in its context.

Visualizing information

Over the last few years, the search model based on a low-level preprocessing of the documents has become the standard approach, even for small collections or single document access. In parallel, text mining tools have often been augmented with visualization techniques, which ensure the browsing functionalities [Hearst, 1999].

Many visualization devices and visual metaphors have been proposed to help users to get an overall picture of the document content and to categorize it. Three simple but interesting principles have emerged from this past experience. The contrast between relevant and irrelevant information must be visually evident (various underlying or size variation means have been used). It is also important that the user can easily pan and zoom the document content. To avoid disorientation, one must have a global view of the document in mind even while focusing on a specific segment. The local and global views are often presented concomitantly and interlinked to guide the browsing. The third important aspect is categorization as it helps the user to distinguish between different types of documents and/or textual segments.

Our approach is based on the same principles.

Natural language processing (NLP)

Even if innovation in text mining has mainly came from visualization techniques in the last decade, we argue that recent advances in specialized language processing can enhance domain specific document access.

Thanks to the development of computational terminology, it is possible to identify the technical vocabulary of a document and the terminological collocations which are often highly semantically relevant in a specialized domain (example "Hypoalphalipoproteinemia"). Terminological analysis can contribute to a more specialized indexing of documents. It has been used to produce explicit indexes of document or collections [Wacholder et al., 2001, Anick, 2001, Nazarenko and Aït El Mekki, 2005].

Current research also focuses on the textual structure of the document. It has been shown that information retrieval can benefit from the exploitation of the structure of the documents (usually represented as an XML markup) [Vittaut and Gallinari, 2006]. It is also clear that extracting a text segment or a specific piece of information must take the context into account: an introduction, a definition or a figure legend do not have the same informational status. It is therefore important to make explicit the structure of the documents [Marcu, 2000, Schilder, 2002].

Finally, the development of methods for corpus-based ontology building [Després and Szulman, 2006] facilitates the creation of ontologies, in which the ontological knowledge is connected with the linguistic one and which can be used for the semantic annotation of documents. The semantic metadata associated with documents can in turn be exploited in text mining, for instance for document categorisation [Pratt et al., 1999].

Navigation model

Relying on our previous work in document annotation [Derivière et al., 2006] and terminological analysis [Aubin and Hamon, 2006], we argue that it is possible to develop new tools for accessing document content, which better addresses the complexity and heterogeneity of users' information needs. We have designed a new model of navigation and a corresponding navigation tool that help the consultation of the medical recommendations by physicians.

As shown above, searching and browsing models to document access both have their own advantages and limitations. Our navigation model aims at taking the best of the two worlds. From usual browsing tools, we

keep the idea that information must always be related to its context. From the standard searching approach, we learn the fact that discriminating between relevant and irrelevant document passages is important. We argue that, compared with traditional approaches, our navigation model proposes a better compromise between quickness, relevance, exhaustivity and reliability.

Evaluating the quality of a navigation

Let $N = (s_{a,1}, s_{b,2}, ..., s_{i,n})$ be a navigation composed of n steps in which the user successively reads the textual segments S_a, S_b.... Some of these segments may be relevant for the user but some others may not. The same segment can be visited twice as the navigation may have cycles or backward steps. A navigation tail is the subsequence of navigation steps that are visited after the last relevant segment has been consulted. The quality of a given navigation is measured according to the following metrics:

- The navigation *precision* is the proportion of relevant segments in the navigation: $precision(N) = r/n$, where r is the number of relevant segments in N.
- The navigation *recall* is the proportion of relevant segments that are retrieved by the system: $recall(N) = r/R$ where R is the total number of relevant segments present in the document.
- The navigation *efficiency* is the proportion of steps that are useful to find relevant textual segments in N: $Eff(N) = (n - rep - tail)/n$, where rep is the number of segment repetitions, and $tail$ the length of the navigation tail.

In an optimal navigation, $recall(N) = precision(N) = Eff(N) = 1$, and $Eff(N)$ is redundant with $recall(N) = 1$. In other cases, however, efficiency distinguishes the case where the user navigates straight to the relevant information ($Eff(N) = 1$) and the case where he/she wanders around (cycles, feedbacks) and/or keep on browsing whereas he/she has found the available information (tail) ($Eff(N) < 1$).

Presenting information in context

According to the visualization principles, we consider that a textual segment must be contextualized, as the context guides its interpretation. Two different localization systems (maps) are proposed to the reader.

The first one is the overall structure of the document (document map), usually represented by its table of contents. It is traditionally explicitly designed by the author of the document to help the reader to locate information. In technical documents such as the medical recommendations, the author's table of contents is often too coarse-grained. NLP document segmentation methods help to identify fine-grained structures in documents. In any case, we suppose that we have an explicit hierarchy of segments, which can be labeled[2] and presented in the form of a tree. The document map is not sufficient however. A document may have several types of readers, whereas the table of contents is designed for a specific reader's profile.

One of the originality of our navigation model is to offer a second localization system (domain conceptual map). It is based on a conceptual model of the domain of the document and oriented by the readers' profile. This conceptual map requires of course that the textual segments are indexed according to the domain conceptual model. In that perspective, the conceptual model can be considered as a segment categorization where a single segment can be attached to different categories. This second map functions as an alternative to the first one. It presents a new organization of the document content. In that perspective, it is similar to a back-of-the book index, which offers a second way to access document, which is complementary to the table of contents. The difference is that the conceptual map is based on the domain model of the document rather than its terminology.

These two maps play an important role in our navigation model. They give a direct access to the textual segments and therefore reduce the navigation length. Compared with full reading or random navigation, they increase both the precision and efficiency.

Discriminating relevant information

The second important feature of our navigation model is the discrimination between relevant and irrelevant textual segments. If the reader is able to express his/her information need in the form of a query, the entire navigation model is parameterized according to that query.

This relevance is based on a terminological analysis of the document which computes a set of variant terms $t_1, t_2, ...t_i$, supposedly semantically equivalent to the user's initial query term t. For instance, if the initial request is "elderly", we consider that a textual segment dealing with "aged patient" is relevant.

For sake of simplicity, lets consider a query q composed of a single term t. We consider the following relevance definitions:

- A textual segment is relevant with respect to q if it contains an occurrence of t or of a variant t' of t. We consider that a segment that contains a relevant segment is not relevant as such but that it nevertheless helps to localize segments.
- A conceptual category is relevant with respect to a query q if one or several of its attached segments are relevant with respect to q.

Once a query has been expressed, the reader is able to discriminate in the document and conceptual maps where are the relevant segments. The segments can be gathered in a single section or scattered all over the document. They can also be clustered under a single cat-

[2]When the segment is an explicit document section, its label is its title. When the segment is smaller such as a discursive frame, its label is given by a "Frame introducer" which is a detached adverbial. Figure 1 shows a frame example introduced by "In a first step".

egory in the conceptual model. This increases the precision/recall of the navigation. The maps help the user to glance at the various contexts in which he/she should look for information. This is important as the navigation must go on until all relevant segments have been read (to augment recall) but should stop as soon as they have been read (to increase efficiency). Our hypothesis is that the maps increase the reliability of the system (which appears as an increased efficiency).

Navigation prototype

Our prototype is currently developed in C++ with the graphical toolkit Qt 4.2. It is fast and portable.

User interface

Four areas are present in our interface, as shown on Figure 2: the keyword input area (area 0), two maps of the document, and the document itself.

The *conceptual map* (area 1) locates the results of the request in a conceptual model. The conceptual model is a hierarchy in which every category used to describe a section of the document is kept. The *document map* (area 2) locates the results of the request in a table of contents. This view gives the user an overview of the structure of the document, such as it was designed by its author. The *document view* (area 3) locates into the document structure the results of the query or the sections which have been selected by the user in one of the other views.

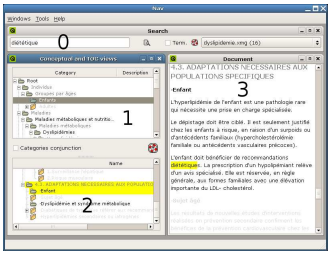

Figure 2: Main window of the user interface

Foreground colors are used to discriminate relevant segments, categories or sections. For instance, a section is light gray if it has no occurrence of the query terms. It is black if it is directly relevant to the query and it is dark gray if the section has a relevant section as subsection. For categories if the conceptual map, the colors are the same but refer to the ratio $prop = rc/tc$ with rc the number of relevant sections attached to the category c and tc the total number of sections attached to c. If $prop < 1/3$ we use light gray, if $prop \geq 1/3$ and $prop < 2/3$ we use gray, and black otherwise. In the document, occurrences of the query terms are highlighted with a yellow background.

Views are linked together. If the user types a keyword, the other views become active: the maps are filled with categories and sections, and the document view shows the entire document. When the user selects one or more categories in the conceptual view, the others are affected: relevant sections are highlighted in the document map and shown in the document view. If the user selects more than one category, the selection depends of the state of the check box named "categories conjunction". If it is enabled, only sections relevant for every selected categories are highlighted. Otherwise, sections relevant for at least one of the selected categories are highlighted. When the user selects some sections in the document map, they are simply shown in the document view. Finally, when the user clicks on the document view, the current section is automatically located in the document map.

Behind the interface

The system is composed of three subsystems. The *user subsystem* relies on the search engine, which locates the occurrences of the query terms in the documents, and computes the relevance of each section. To improve the accuracy, the search engine exploits a terminology which contains a list of terms and their synonyms. A tree deployment module chooses dynamically how the hierarchies of conceptual and document maps are deployed, expanding relevant nodes, and to collapsing others. The *knowledge management subsystem* contains a tagged corpus, a domain model used to type the corpus and a set of terminological relations. The *administrator subsystem* is composed of NLP tools used for knowledge acquisition. Our prototype relies on four types of tools for terminology acquisition[3], corpus-based ontology building [Després and Szulman, 2006], corpus preparation (conversion, segmentation) and corpus annotation.

Navigation scenarii

The user can choose to navigate with the conceptual map, with the document map, through the document itself or by mixing different views. The selection of categories can be viewed as a semantic query expansion. For example, if the initial request is "dietetic" and the selected category is "young people" then the user wants to view every part of the text dealing with young people and containing the keyword 'dietetic' (or one of its variations). When the user chooses more than one category, it could be a conjunction or a disjunction of each, depending of the state of the check box described above.

Using the document map, the user may have a direct access to any section identified as relevant for the query. It is useful when information is scattered or to vizualize the repartition of a specific term (*e.g.* a molecule name) through the document.

The document view shows the graphical aspect of the document, or its length. Since each section is coloured with respect of the relevance, the user has indications to navigate in the document with the scroll bar.

The recommended navigation strategy exploits the different views. The user enters a keyword, then select some categories to expand its request, uses the document map to jump in relevant sections and read them in the document view.

[3]The current version of the protype is based on YaTeA [Aubin and Hamon, 2006] and Faster [Jacquemin, 1995].

Discussion: search *vs.* navigation

Our navigation prototype has not been evaluated yet. This section presents detailed examples of navigation, which show that the navigation model seems to be more efficient than the classical search method. The end of the section also present the evaluation protocol that we plan to set up to validate the proposed approach.

We have defined with medicine experts (LIM&BIO, Paris 13) a sample of queries which are relevant for a physician in consultation. One of those is the following: "What treatment should I prescribe to my elderly patient who has no cardiovascular disease history and who has been on a diet with no success ?" We compare how the user can find the relevant textual segments using either our navigation prototype or a classical search engine. If we suppose that the physician chooses the keyword "elderly" in order to formulate his query (currently our system only handles single keyword queries), both systems return the same 20 textual segments (those containing the keyword) but the results are presented in different ways as shown on Figures 3 and 4.

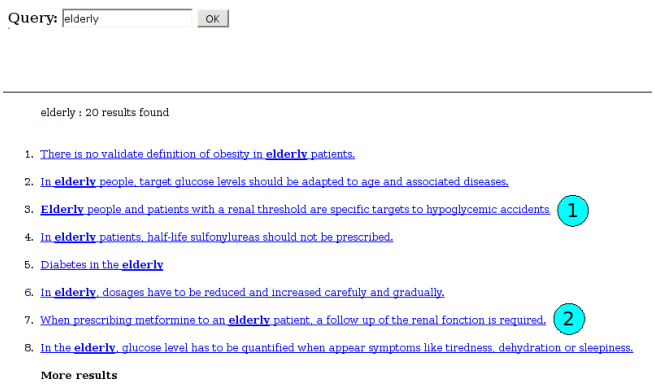

Figure 3: Result presentation (classical search method)

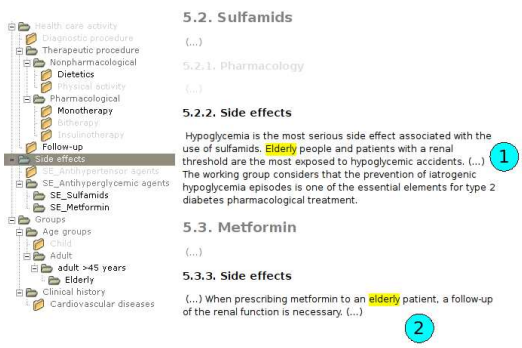

Figure 4: Result presentation (navigation model)

The search method presents the result list, with the sentences containing the keyword occurrences, whereas the navigation method presents the results with respect to the document and the conceptual maps[4]. In our approach, categories associated to one or more result (like "follow-up" or "side effects") are coloured differently from categories which are not associated with any result (like "bitherapy" or "physical activity"). We want to show how these different result presentations affect the navigation quality with respect to our three criteria: precision, recall and efficiency.

Example 1 Let's consider the following result (number 1 in the two figures):

- elderly *people and patients with a renal threshold are a specific target to hypoglycemic accidents*

In the search method, the meaning of this result cannot be fully captured because the sentence is not a sufficient context. If we read a larger segment, we understand that these accidents are side effects of a drug class which specifically affects elderly patients. The navigation model gives more contextual information. The keyword (*i.e.*"elderly") is presented in a larger textual segment than the sentence, which facilitates the comprehension. The result is also localized on the conceptual map, where it falls under the category "Side effects". This givee the context required to get the whole meaning of the result and consequently increases the user ability to quickly discriminate relevant results from irrelevant ones.

Example 2 Let's now consider the two following results (numbers 1 and 2 in the figures):

- elderly *people and patients with a renal threshold are a specific target to hypoglycemic accidents*
- *where prescribing metformine to an* elderly *patient, a follow-up of the renal function is necessary*

With the list presentation of a traditional search engine, the two results are disconnected whereas they both deal with drugs side effects that specifically affect elderly people. In contrast, in the navigation model, these two results are linked together because they fall under the same category "Side effects". We argue that the navigation model facilitates the user understanding and thus increases the navigation precision and efficiency.

Example 3 With the search method, if the users want to be sure that they have got all the results needed, they have to browse all the items until the last one even if they have already got all the relevant ones. The navigation model is designed to avoid this useless browsing by showing a global, synthetic and well-comprehensive view on the results through the conceptual map.

For instance, a physician will be able to quickly discriminate on the map the relevant categories like "Elderly people", "Pharmacological treatment" and "Side effects" and the irrelevant ones like "Follow up" (because he/she is interested in first treatments, for instance) or "Cardiovascular disease history" (because

[4]Only one map is presented on Figure 4 for sake of lisibility.

his/her patient has no history of that type). So, after he/she has read the results in the relevant categories, he/she can stop browsing without reading the remaining results. In that way also, the navigation model increases physician's efficiency.

These three examples show that the navigation model offers to physicians a better control of the research process and therefore increases the navigation efficiency.

Evaluation protocol Even if this navigation model is based on commonsense principles and if it seems to perform properly on some navigation examples, it must be more thoroughly evaluated.

To analyze the advantages and drawbacks of our navigation model, we plan to compare it with other traditional methods used to get access to the document content, as shown in the examples above (manual browsing, traditional information retrieval and search assisted by a table of content or a back of the book index). The various methods will be compared on formal grounds. We are currently defining a unified language L to describe the different document access methods in the form of a list of the browsed titles and segments.

We have defined a query test set with our physician partners. For the first evaluation, few physicians will be asked to answer the test queries using the different document access methods, while an independent observer will codify their various operations in L. The various methods will be then compared on the basis of their resulting document access traces (length of the navigation trace, presence of cycles, length of tails, etc.).

Conclusion

We have presented a new navigation model that combines the search and the browsing techniques to give to the user a comprehensive and flexible tool for accessing document content. Our approach relies on a terminological relevance calculus and on two different maps, which help the user to localize any textual segment(s) in the whole document and with respect to the conceptual model of the domain. The dynamicity of our system ensures that the coloration of the maps and the document is automatically updated according to any new user's query. The user sees what is relevant but remains free to navigate as he/she wants.

The analysis of a small set of navigation log confirms our initial intuitions but our model and prototype must be more thoroughly evaluated. We plan to compare our navigation model with other existing models (manual browsing, traditional information retrieval and search assisted by a table of content or a back of the book index).

References

Anick, P. G. (2001). The automatic construction of faceted terminological feedback for interactive document retrieval. In Bourgault, D. *et al.* editors, *Recent Advances in Computational Terminology*, pp. 29–52. John Benjamins, Amsterdam.

Aubin, S. and Hamon, T. (2006). Improving Term Extraction with terminological resources. In Salakoski, T. *et al.* editors. *Advances in Natural Language Processing*, LNAI 4139, pp. 380–387. Springer.

Baeza-Yates, R. and Ribeiro-Neto, B., editors (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Company, Wokingham, UK.

Burger, J. and al (2002). Issues, tasks and program structures to roadmap research in question & answering (q&a). Technical report, DARPA.

Chi, E., Hong, L., Heiser, J., and Card, S. (2004). ebooks with indexes that reorganize conceptually. In *Proc. of the Human Factors in Computing Systems Conf.*, pp. 1223–1226, Vienna. ACM Press.

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer*, 20(9):17–41.

Derivière, J., Hamon, T., and Nazarenko, A. (2006). A scalable and distributed NLP architecture for Web document annotation. In Salakoski, T. *et al.* editors, *Advances in Natural Language Processing*, LNAI 4139, pp. 56–67. Springer.

Després, S. and Szulman, S. (2006). Terminae method and integration process for legal ontology building. In *Advances in Applied Artificial Intelligence*, pp. 1014–1023. Springer Berlin/Heidelberg.

Hearst, M. A. (1999). *Modern Information Retrieval*, chapter User Interfaces and Visualization. Addison-Wesley Longman Publishing Co., Wokingham, UK.

Jacquemin, C. (1995). A symbolic and surgical acquisition of terms through variation. In *Learning for Natural Language Processing*, Lecture Notes in Computer Science, pp. 425–438.

Marcu, D. (2000). The rhetorical parsing of unrestricted texts: a surface-based approach. *Computational Linguistics*, 26(3):395–448.

Nazarenko, A. and Aït El Mekki, T. (2005). Building back-of-the-book indexes. *Terminology*, 11(1):193–218.

O'Hara, K. (1996). Towards a typology of reading goals. Tech. Report EPC-1997-107, RXRC/Cambridge Lab., Cambridge, UK.

Pratt, W., Hearst, M. A., and Fagan, L. M. (1999). A knowledge-based approach to organizing retrieved documents. In *Proc. of the 16th Nat. Conf. on Artificial Intelligence*, pp. 80–85.

Schilder, F. (2002). Robust discourse parsing via discourse markers, topicality and position. *Natural Language Engineering*, 8(3):235–255.

Vittaut, J.-N. and Gallinari, P. (2006). Machine learning ranking for structured information retrieval. In *Proc. of the European Conf. on Information Retrieval*, pp. 338–349.

Wacholder, N., Evans, D., and Klavans, J. (2001). Automatic identification and organization of index terms for interactive browsing. In *Proc. of First ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 126–134, Roanoke, VA.

Zargayouna, H., Aït El Mekki, T., Audibert, L., and Nazarenko, A. (2006). IndDoc: An aid for the back-of-the-book indexer. *The Indexer*, 25(2):122–125.