Intelligent Clustering as a Source of Knowledge for a Web Dialogue Manager in an Information Retrieval System

Paulo Quaresma and Irene Pimenta Rodrigues
Departamento de Informática, Universidade de Évora,
Largo dos Colegiais, 7000 Évora, Portugal
{pq|ipr}@di.uevora.pt

Abstract We present a dialogue manager for a Web information retrieval system that uses intelligent clustering techniques in order to be more cooperative. The proposed system provides an intelligent behavior during user interactions through the use of domain knowledge and the construction of an interaction context. Domain Knowledge is used to build clusters of documents which are presented to the users as graph structure and as choice menus. The interaction context is used to allow the system to interpret a new query in the context of the previous interactions. We present a detailed example of an interaction with the proposed system.

Keywords: Information Retrieval, Intelligent clustering, Web dialogue system

1 Introduction

In IR dialogues the user wants to look for some documents and his queries are ways of selecting sets of documents; the system questions and answers always intends to help the user in his search of documents by supplying information on subsets of documents in the text database.

The main problem with our goal is the huge amount of knowledge necessary for the semantic interpretation of the user queries (in natural language sentences or in a sql like syntax or from choice menus). Since it was not reasonable to manually build such a large knowledge base, covering all the subjects of all documents, we decided to study the possibility of automatically extract some knowledge from the texts. This knowledge can be used in the interpretation of user queries in order to provide useful answers even when the system is unable to understand the full meaning of the queries.

One of the ways to extract knowledge from a large documents database is by using the results of grouping documents into topically-coherent groups

(clusters) with topical terms that characterize them. The clustering and reclustering algorithms must be done on-the-fly, so that different topics are seen depending on the set of documents selected by the user queries.

We developed algorithms that are able to select the more relevant clusters from the set of all existed clusters. These clusters are the more informative ones, with relevant topics that may completely cover a sub-collection topics. These algorithms allow us a guided search in the state space of possible clusters using admissible heuristics.

In order to extract knowledge from the documents database we decided to cluster documents based on two different characteristics: citations and subjects. All documents were previously analyzed and a database relating each document with its citations and subjects was built. Then, for each user query, a set of documents is obtained (using an information retrieval engine) and these documents are clustered using the relations previously calculated.

The system has a Dialogue manager that uses the results of clustering a sub-collection of documents and the interaction context in order to supply the user pertinent information for further refinement.

The dialogue manager cooperatively helps users in their searches by supplying an appropriated set of topics and more relevant documents that characterize the collection of documents selected by the user query.

Section 2 describes the documents classification process accordingly with a set of concepts previously defined by the Portuguese Attorney General Office (PAG Office). Section 3 presents the different clustering methods: clustering by citations and clustering by topics using the information that results from the documents classification. Section 4 presents the dialogue manager that is responsible for helping a user defining his goal. It uses: the user interventions, they provide information on user intentions; and knowledge of the text database. This knowledge is obtained from different ways using the results of the clustering processes and domain rules in a knowledge base. Section 5 presents the system architecture implemented in a Linux environment using XSB Prolog. And finally in section 6 we present a detailed example of an interaction with our system.

2 Document Classification

It is very important that all documents are classified accordingly with a juridical taxonomy. In fact this classification is the basis of the clustering mechanisms.

The juridical taxonomy is a Juridical thesaurus that is a result from the project: PGR - Selective access of documents from the Portuguese Attorney General.

The juridical terms thesaurus can be described as a taxonomy which has the relations:

is equivalent to ex: law is equivalent to norm; *is generalized by* ex: prime minister is generalized by minister; *is specified by* ex: accident is specified by traffic accident; *is related with* ex: desertion is related with traffic accident.

The thesaurus is also used to expand queries to include all the values that are equivalent or more specific or related, with the initial query (more information can be found in [15]).

Our information retrieval system is based on a legal database composed by texts from the Portuguese Attorney General.

However, some of the texts did not have a juridical analysis field, i.e., they were not previously classified accordingly with a juridical taxonomy. In order to handle this situation, we have developed an automatic juridical classifier based on a neural network. The classifier receives as input a legal text and proposes a set of juridical terms that characterise it. The proposed terms belong to the taxonomy of juridical concepts developed by the Portuguese Attorney General Office.

2.1 Automatic Classification

The classifier was developed using the Stuttgart Neural Network Simulator [17] and the network is a feed-forward network with two layers of units fully connected between them. The first layer has the input units and the second layer has the output units. Each input unit is associated with a specific word and its input value is the word frequency in the text. Each output unit is associated with a juridical term and its value is 1 or 0, defining if the juridical term characterises the input text.

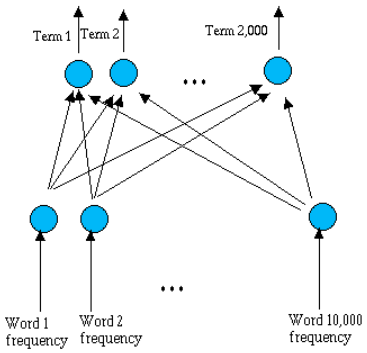
In order to build the network it was necessary to create a mapping between the text words and the input units. After analysing the legal texts we obtained a set of 10,000 distinct words, composed only by nouns, verbs, adverbs, and adjectives. In this process we have discarded all the other word classes and we have reduced each word to its canonical form (infinitive for verbs, singular for nouns). We have also mapped each used juridical term (2,000) to a specific output unit. Finally, connections between all input units and all output units were created: 10,000 x 2,000 = 20,000,000.

The figure 2.1 shows the network topology.

As learning algorithm for this feed-forward neural network, it was used the standard backpropagation algorithm and the net was trained until the SSE (sum of squared errors) was less than 0.1.

The training set was composed by 95% of the texts from the Portuguese Attorney General and the validation set was composed by the other 5%. We also have a test set composed by other legal texts not previously classified.

As results for the validation set we obtained that 80% of the proposed terms were correct. It is possible that the other terms are not completely incorrect. In fact they may be a different characterization of the text and they have to be analyzed by juridical experts.



3 Clustering

Clustering is a complex process [16] since it involves: the choice of a representation for the documents, a function for associating documents (measures for similarity of documents with the query or between them) and a method with an algorithm to build the clusters. One of the best clustering methods is the Scatter/Gather browsing paradigm [4, 5, 8] that clusters documents into topically-coherent groups. It is able to present descriptive textual summaries that are built with topical terms that characterise the clusters. The clustering and reclustering can be done on-the-fly, so that different topics are seen depending on the subcollection clustered.

In our framework we decided to cluster documents based on two different characteristics: citations and topics or subjects. All documents were previously analyzed and a database relating each document with its citations and subjects was built.

Then, for each user query, a set of documents is obtained (using an information retrieval engine) and these documents are clustered using the relations previously calculated. Finally, the obtained clusters are visualized as a list or as a graph structure (using the Graphviz package developed by AT&T and Lucent Bell Labs).

3.1 Clustering by subject

In order to obtain the clusters of topic relations it was necessary to classify each document accordingly with a set of concepts previously defined by the Portuguese Attorney General Office (PAG Office). The classification was done manually by the PAG Office and automatically by a neural network.

The documents were parsed in order to build the topics relationship database.

Using this database it is possible to obtain several graphs with topic relations that can give rise to several lists of clustered concepts.

3.2 Graph of topic relations

One way of building the relations graph is by creating a graph arc between a pair of documents when there are at least 90% of common topics. As an example, the query "bombeiro" obtains the following graph of topic relations (see fig 1, that was build using the Graphviz package developed by AT&T and Lucent Bell Labs).

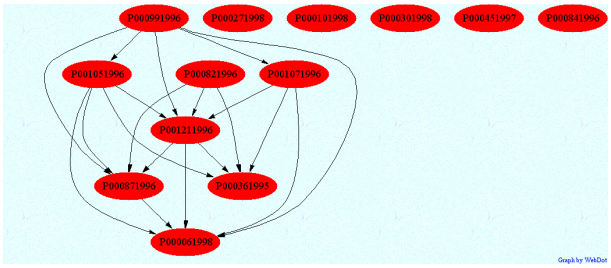


Fig. 1. Topics: Bombeiro – Fireman

In this figure is possible to detect a cluster of closely related documents and a set of non related documents (probably about some minor distinct subjects). This procedure, selecting the clusters, is done automatically using an algorithm that is able to select the topics for characterizing the clusters.

Different relations graphs may be build by using different criteria when deciding to create a graph arc between a pair of documents.

Topic Clustering As it was already described, it was built a database relating each document with the set of their topics. After a user query this structure is transformed in another structure that associates each topic with the set of the retrieved documents with this concept. These structures are shown in figure 2.

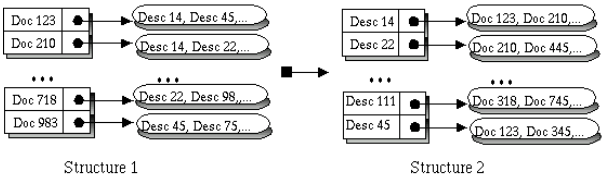


Fig. 2. Document structure

Finally we must must choose a set of topics such that:

1. The union of the set of documents associated to the topics is the initial set of documents.
2. The intersection of the set of documents associated to any two topics is empty.

These two conditions can not be satisfied always. Whenever this happens the first one is dropped.

However there are other proprieties the set of topics should have:

1. Its cardinality should be between 10 and 20.
2. The cardinality of each document set should be similar.
3. Descriptors with only one document associated should be ignored.

Since it is not possible to search all the state space in a reasonable time we had to use some heuristics in order to cut off part of the search space, and we used an informed search algorithm, a best first search with an evaluation function specially designed for this problem.

This procedure starts by:

- Sorting structure 2 by descendent order of the cardinality of the documents set.
- Eliminating the topics with only one document associated.

Then the best first search is guided by an evaluation function that always choose to add a topic that as a set of documents with its cardinal as near as possible of the interval [10, 20].

The search ends with success when:

- All documents are selected, the union of the sets associated with the selected topic is the set of selected documents.
- The cardinal of the set of topics reaches 30, and the cardinal of the union of the sets of documents is greater then 70% of the initial number of documents.

Evaluations of this algorithm guarantee that it will take O(n*m), n is the number of documents, m is the number of topics in structure 2 without those eliminated in the first step. For 10000 documents and 2000 topics it takes about 100 milliseconds, a reasonable time for a search in the World Wide Web.

3.3 Clustering by Citations

is obtained by processing complete set of legal documents from the Portuguese Attorney General to compute the citations between documents.

In order to obtain all the citations it is necessary to construct a specialized chart parser, which is able to partially analyze the text and to retrieve and to normalize the citations between documents. Note that documents can be

cited by their number, date, title, author, and by almost any mixture of these fields. Taking this fact into account, we have built a database with these fields for each document and, whenever a possible citation appears in the text, the parser checks if the cited document exists and it builds a new citation entry in the database.

These citations are used by the system to build the graph of relations between the set of documents retrieved by the user queries.

As an example, the query "bombeiro" (fireman) obtains the graph of citations of fig 3.

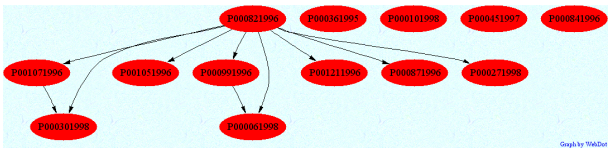


Fig. 3. Citations: Bombeiro – Fireman

Note that it is possible to detect an important legal document that is referred by most of the other retrieved texts. It is, a document that created jurisprudence on fireman cases.

Using this information the system is able to supply a set of relevant documents with the appropriated topics to the user. In the law field this information is important and useful.

4 Dialogue Manager

Dialogues in the context of information retrieval systems are different from traditional dialogues in computational linguistics [9, 14, 3, 2, 10] because our user normally does not have a plan to execute, he actually does not have a precise goal such as: 'go to Boston' or 'phone to Mary'. Our users may want to see some documents, but they do not know which particular documents.

The function of our dialogue system is to help a user defining his goal. In order to accomplish this purpose our system uses: the user interventions, they provide information on user intentions; and knowledge of the text database, this knowledge is obtained from the results of the clustering processes and from domain rules in a knowledge base.

In traditional dialogues the system must recognize the user goals and his plans to achieve them[9, 14, 3]. The system represents and infers the user intentions using domain knowledge from a library of plans and checking if a plan is correct. The discourse structure of this kind of dialogues is complex, it has many kinds of segments: continuation, elaboration, repair, clarification, etc.; and their structure is built through the inference of user intentions,

taking into account clue words and using plans structure (task and dialogue plans)[7, 6, 12, 10, 2]. In our system each event (utterance) is represented by logic programming facts that are used to dynamically update the previous model [1]. Using this approach it is possible to represent new events as update logic programs and to obtain the new states. Moreover it is possible to reason about past events and to represent non-monotonic behaviour rules. Each utterance will trigger the inference of the user intentions taking into account the user attitudes (such as his beliefs and the user profile). The results of the inference of the user intentions are:

- a new set of user and system beliefs
- a new set of user and system intentions (such as the intention of the user to be informed of something by the system)
- a new dialogue structure. This structure keeps the dialogue context allowing for the interpretation of user acts in its occurrence context. The dialogue structure constraints the interpretation of user intentions and it is built as a result of the intentions inference.

In our dialogues the user wants to look for some documents and his queries are ways of selecting sets of documents; the system questions and answers always intends to help the user in his search of documents by supplying information on subsets of documents in the text database.

After a user query the system may:

Show the set of documents selected by the query.

Since our information retrieval system is boolean, the documents that are selected are just those that match the query.

Show the set of documents selected by the expanded query.

Our information retrieval system has options for expanding a query such as:

- expand using morphologic flexion: verbs, nouns, adjectives, etc.;
- expand using synonyms (a general dictionary);
- expand using a domain thesaurus.

Present a set of keywords that may help the user to refine his query.

In order to build a set of keywords the system may build groups of documents (clusters) from the initial set selected by the user query. These groups of documents are disjoint, i.e. there are no document that belongs to more then one group.

Present a set of concepts that may help the user to refine his query.

In cases where the system has knowledge about some of the documents subject it is possible to build groups of documents using that knowledge, and to provide the user concepts for refining its query.

Explain the user why his query does not select any document, providing suggestions for other queries.

Most information retrieval system assume that the user will never get to a dead-end with its queries, so they relax on the meaning of a query by using non boolean retrieval systems.

These systems are appropriate for general Web searches, but for a system that intends to control the search we think that a boolean system is best suited.

A user query must be interpreted in context. It is through the inference of user intentions that our system interprets the user query in its context.

The system takes into account user profiles in the inference of user intentions. Since our database has juridical documents, we have different profiles for lawyers, law students and other users.

5 System Architecture

In order to build the dialogue manager for a web information retrieval system we must take into account that:

- The number of users registered in the system is large (thousands).
- Typically there are more then one user using the system at the same instant.
- The users may interrupt their session for large periods of time.
- The users would like to be informed when a previous query result changes due to updates in the database.

These facts make impossible to use a process to deal with each user because the number of processes in an operative system is limited, and most of the processes will be blocked waiting for a user request.

The architecture that we designed to solve this problem has:

- An agent manager that receives all the user requests and keeps a database with all the users interrogation (dialogue) context.
  The agent manager is implemented in Prolog: it interfaces the web user requests; it verifies if the user is registered and if it has no pending requests; then it launches other Prolog agent, the agent process.
- Several process agents that given an user request and its interrogation context are able to answer the respective user and to actualise his interrogation context.
  The agent process when it is launched: consults the user database; answers the user; updates the user database (interrogation context); and informs the agent manager that it has finished.
- An agent monitor that informs the agent manager of the latest changes in the documents databases. These changes are transmitted to all users that have one of their previous queries results changed.
  This Prolog agent consults all the users database to check for differences in the user query results. When there are changes in a user, the agent adds a new request in the user database that will be handled by the agent process.

To model the knowledge the agent process represents four levels of knowledge using extended logic programming. The knowledge levels are: Interaction, Domain, Information Retrieval and Text.

The Interaction Level is responsible for the dialogue management. This includes the ability of the system to infer user intentions and attitudes and the ability to represent the dialogue sentences in a dialogue structure. The dialogue structure will be kept in the user database.

The Domain Level includes knowledge about the text domain and it has rules encoding that knowledge.

The Information Retrieval Level includes knowledge about documents and their relationship. The agent process has logic programming rules for computing labelled clusters that will be used as knowledge about the documents selected by the user.

The Text Level has knowledge about the words and sequence of words in each text of the knowledge base.

The architecture is based on three specialised Prolog agents:

- An agent manager that receives all the user requests and it acts as an interface with the specific user process agent;
- The user process agent which is specific to each user and it has information about the user profile and the previous interrogation context;
- An agent monitor that informs the agent manager of the latest changes in the documents databases allowing these changes to be transmitted to all users that have one of their previous queries results changed.

As it can be seen in figure 4, each user communicates with the agent manager, which redirects the event to the specific user process agent (launching the user process, if needed). In order to obtain a cooperative answer, each user process agent accesses the respective user interaction context (composed by logic programming facts) and the databases. Afterwards, it updates the interaction structure and communicates the answer to the agent manager, which redirects it to the user.

On the other hand, the monitor agent is always accessing the databases and the interaction structures trying to detect changes in the given answers. Then, it informs the agent manager, which will inform the change to the user process agent.

This architecture was implemented in a Linux environment using XSB Prolog.

6 Interaction Example

Suppose the user wants to find documents about militaries that have been injured and received a pension from the government. He wants to find out if there have been similar cases in the legal knowledge base of the Portuguese Attorney General.

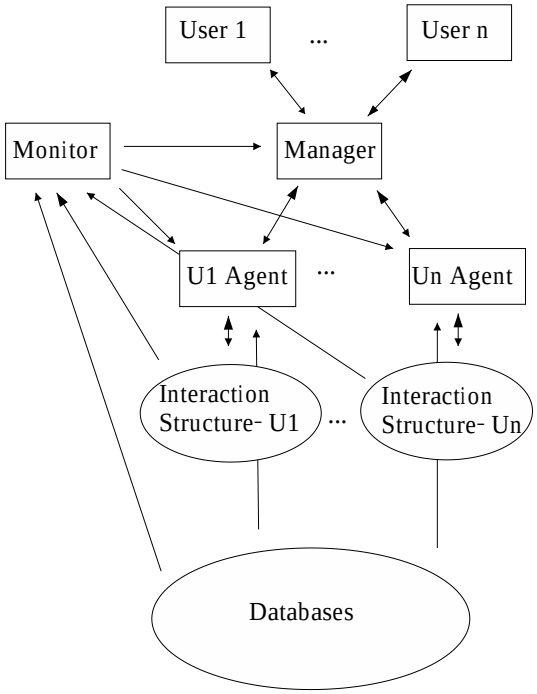


Fig. 4. General architecture

In order to help readers and to keep the example short, the Portuguese web version is not presented. Instead an English compacted version of the interaction is shown. The complete web interaction can be easily reproduced from the URL http://www.pgr.pt

[User - Q1:] Militares (militaries)

[System - Q2:] Result: 1127 documents

Documents with keyword:
(120) incapacity
(86) medal
(28) court
(16) commission
(11) rights
(4) marriage
...

Options on this list were obtained by clustering the set of 1127 documents.
[User - Q3:] Chooses: incapacity
[System - Q4:] Result: 120 documents

Documents with keyword:
(10) service
(6) pension
...

Options in this column are obtained by re-clustering the 120 documents.
[User - Q5:] Chooses: pension
[System - Q6:] Result: 6 documents

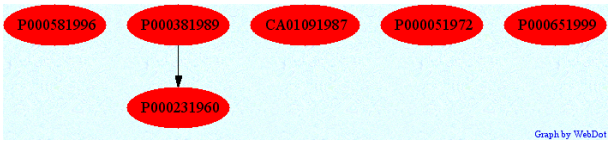


Fig. 5. Graph clustering by topic

Options in figure 5 were obtained by graph clustering by topic the 6 documents.

[User - Q7:] Refinement: illness (after analyzing the topics of the retrieved documents and deciding that they were not relevant for his goals)

[System - Q8:] There are no documents of the 6 above that match your new constraint, I assume that you want your query to refine Q3, i.e. you are looking for documents about: militaries and incapacity.

[User - Q9:] Result: 6 documents

Options in figure 6 were obtained by graph clustering by topics the 6 documents.

.....

This example shows some of the flexibility and cooperativeness of the system, allowing users to dynamically refine their queries and helping them in a pro-active way, giving hints and clustering the retrieved documents.

During the interaction, the tree representation of the dialogue is being inferred and displayed in a visual tree-diagram. As an example, the tree

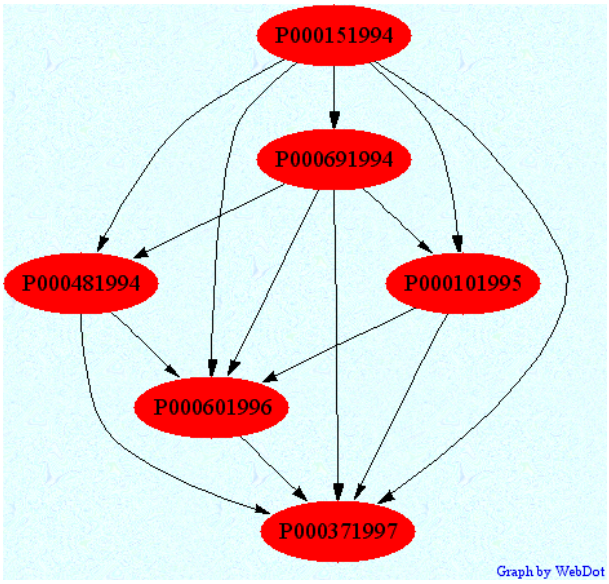


Fig. 6. Graph clustering by topic

representation of the previous example is presented in figure 7 (in a compact version).

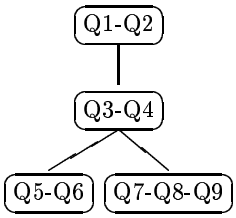


Fig. 7. Interaction Structure Tree

7 Conclusions

We have used the results of documents clustering as a source of knowledge for a Web Dialogue Manager. As we have presented the result of our option seems to be adequate since it gives us a way to obtain general and specific knowledge in order to enable the system to infer and fulfill the user intentions.

The evaluation of our system has not been accomplished in a formal way. The quantitative evaluation of this system is hard to perform and by now we do not have done it. A qualitative evaluation can be done by taking into account the system logs and user comments.

By analyzing the system logs we can conclude:

- Most queries (90%) are done using the multimodal interface. Most users do not use the natural language interface, they prefer to use choice menus, or to use free text queries (keywords with boolean connections).
- The interaction context is frequently used by our users (on average twice on each session). The users use it in order to return to a previous interaction point.
- The system suggestions for query refinement are used in 90% of the cases.
- Most of the system suggestions (70%) are obtained using the information retrieval level.

As for the portability of our dialogue system into other domains document database, we have not tryed to do it yet, but the main issues are:

- A robust natural language grammar enabling us to always obtain the speech act associated to a user multimodal act (it may involve adding some vocabulary and some knowledge representation rules, mainly a domain thesaurus to expand the users queries).
- A knowledge base modeling some domain knowledge. If the system does not have this domain rules it still can act by giving suggestions computed using the knowledge obtained trough the document clustering.
- The computation on-the-fly of documents clusters with a topical expression associated with each cluster.
  The algorithms that we have used are independent from the domain knowledge, so this will not be an issue. The problem will be to obtain the automatic documents classification for these algorithms.

References

1. J. J. Alferes, J. Leite, L. M. Pereira, H. Przymusinska, and T. Przymuzinski. Dynamic logic programming. In *Proc. of KR'98*, 1998.
2. Sandra Carberry and Lynn Lambert. A process model for recognizing communicative acts and modeling negotiation subdialogs. *Computational Linguistics*, 25(1), 1999.
3. J. Chu-Carroll and S. Carberry. Response generation in planning dialogues. *Computational Linguistics*, 24(3), 1998.
4. D. R. Cutting, J. O. Pedersen D. R. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. 15th Annual Int'l ACM SIGIR Conf. on R&D in IR*, June 1992.
5. D. R. Cutting, D. Karger, and J. Pedersen. Constant interaction-time scatter/gathern browsing of very large document collections. In *Proc. of the 16th Annual Int. ACM/SIGIR Conf.*, Pittsburgh, PA, 1993.

6. B. Grosz and S. Kraus. Colloborative plans for complex group actions. *Artificial Intelligence*, 86(2), 1996.
7. Barbara Grosz and Candice Sidner. Attention, intentions and the structure of discourse. *Computational Linguistics*, 12(3), 1986.
8. M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis:scatter/gather on retrieval results. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference*, Zurich, June 1996.
9. Diane Litman and James Allen. A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11(1), 1987.
10. Karen E. Lochbaum. A collaborative planning model of intentional structure. *Computational Linguistics*, 24(4), 1998.
11. Brian Logan and Karen Sparck Jones. Belif revision and dialogue management in information retrieva. Technical report, University of Cambridge, Computer Laboratory, 1994.
12. S. McRoy and G. Hirst. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics*, 21(4), 1995.
13. Johanna Moore and Cecile Paris. Planning text for advisory dialogues:capturing intentional and rhetorical information. *Computational Linguistics*, 19(4), 1993.
14. Martha Pollack. Plans as complex mental attitudes. In Philip Cohen, Jerry Morgan, and Martha Pollack, editors, *Intentions in Communications*. MIT Press Cambridge, 1990.
15. P. Quaresma and I. Rodrigues. Pgr: A cooperative legal ir system on the web. In Graham Greenleaf and Andrew Mowbray, editors, *2nd AustLII Conference on Law and Internet*, Sydney, Australia, 1999. Invited paper.
16. Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989. Reading, MA.
17. University of Stuttgart. *SNNS - Stuttgart Neural Network Simulator*, 1998.