A Dialogue Manager for a WWW-Based Information Retrieval System

Paulo Quaresma and Irene Pimenta Rodrigues

Departamento de Informática,
Universidade de Évora,
Portugal
{pq|ipr}@di.uevora.pt

Abstract. We present a collaborative dialogue framework for web information retrieval systems. The dialogue manager, whose main role is to help users in their documents searches, infers the user and system intentions, represents the interaction context and behaves as an intelligent interface between the user and the information retrieval system. The knowledge used by the dialogue manager in the inference and fulfilment of user and system intentions is mainly obtained trough the intelligent clustering of the documents in the database. The system also uses a knowledge base with some rules modelling domain knowledge on the documents subject (in our application, juridical knowledge). A detailed example in the law field is presented.

1 Introduction

We aimed to include a natural language interface in an information retrieval system. We had a text database with juridical documents that were indexed with SINO, a boolean text search engine from the AustLII Institute [5], which allows for SQL like queries on words and expressions in the documents.

In order to perform a dialog with the user, we want:

— To infer what are the user intentions with the queries. When a user asks for documents with a particular keyword, usually he is interested in documents that may not have that keyword and he is not interested in all documents with that keyword.

— To supply pertinent answers or questions as a reply to a user question. The system must supply some information on the set of documents selected by the query in order to help the user in the refinement of his query.

And we need:

— To record the previous user interaction with the system (user questions and the system answers). This record will play the role of a dialogue structure. It provides the context of sentences (questions and answers), allowing the system to solve some discourse phenomena such as anaphoras and ellipses (solving them in user interventions and generating them in system interventions). Since our system is multi-modal, other user acts such as button clicks and menu choices are also represented in our dialogue structure.

— To obtain new partitions (clusters) of the set of documents that the user selected with his query(ies).

— To use domain knowledge whenever the system has it.

Dialogues in the context of information retrieval systems are different from traditional dialogues in computational linguistics [9, 13, 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 different ways using the same sources (documents) and domain rules in a knowledge base. In traditional dialogues the system must recognize the user goals and his plans to achieve them[9, 13, 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, 11, 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.

To model the knowledge about the documents in the texts database the system represents four levels of knowledge using dynamic 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 in order to obtain the semantic representation of the dialogue. The domain level includes knowledge about the text domain and it has rules encoding that knowledge. For instance, in the law field it is necessary to represent under which conditions a pension for relevant services may be given to someone; those pensions are usually attributed to militaries or to civilians such as firemen, doctors, and nurses. The Information Retrieval Level includes knowledge about what we should expect to find in texts about a subject, for instance that in texts about pensions for relevant services, the pension may be attributed or refused. The Text Level has knowledge about the words and sequence of words that are in each text of the knowledge base.

2 Dialogues

As we stated in the introduction our system dialogues are different from traditional dialogues in computational linguistics literature, because we do not have a plan library to check if the user goals or plans match some plan scheme. In our kind of dialogues it is difficult to build such a plan library to let the system conduct the dialogue by trying to recognizing the user plans and goals.

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.

Dialogue 1:

Example of a dialogue with a user that has profile *other users*. Suppose that the user is a retired fireman that has saved a child from drowning in a river during his holidays putting his life at risk. Since he has caught a pneumonia as a consequence, he wants to know if he can get a pension for relevant services to his country.

User - Q1: Documents about pensions for relevant services?

System - Q2: There are 243 documents, listed in the right frame. For further refinement you may choose one (or none) option in each column:

Pensions that were:	*Pensions for:*	*Documents with keyword:*
given	militaries	drowning
or refused	or civilians	or medal
		or in duty
		or ...

User - Q3: chooses: *civilians* and *drowning*.

System - Q4: There are 45 documents, listed in the right frame. For further refinement you may choose one (or none) option in each column:

Documents with keyword:
death
or accident
or ...

User - Q5: Documents where there was a drowning in a river.

System - Q6: Since there are 13 documents of the 45 above that match the concept, I've listed them in the usual frame. For further refinement you may choose one (or none) option in each column:

Documents with keyword:
car crash
or bridge
or ...

User - Q7: Where there was someone swimming?

System - Q8: Since there are no documents of the 13 above that match your new constraint. I assume that you want that your query refines

Q3, i.e. you are looking for documents with the concept swimming, and drowning and pensions for relevant services for civilians. Do you want to see these documents?

User - Q9: Yes.

3 Dialogue structure

The system builds the dialogue structure to record both user and system questions and answers. This structure is used to compute the meaning of an user query and to allow the user to return to a previous point of the dialogue and to build a new branch from there.

The Dialogue structure is made of segments that group sets of sentences (user and system sentences). The dialogue structure reflects the user intentions, it is built by taking into account the user and system intentions. The dialogue segments have precise inheritance rules defining how segments heritage their attributes from the attributes of their sentences.

The dialogue structure also enables the system to recognize and to generate pertinent discourse phenomena such as anaphoric references.

In this paper we present the following 4 segments that enable us to build a dialogue structure:

Basic — has 2 arguments: Speaker;Sentence Semantic Representation

New — has 2 arguments: Dialogue Structure; Dialogue Structure. This Dialogue Structure inherits their attributes from their second argument.

Ex: New([],basic(User,Q1))

Specify — has 2 arguments: Dialogue Structure; Dialogue Structure. This Dialogue structure inherits their attributes from both dialogues structure.

Ex: Specify(Basic(User,Q1),Basic(System, Q2))

Repair — has 2 arguments:Dialogue Structure; Dialogue Structure. This Dialogue structure inherits their attributes from the second dialogue structure

We may consider that Dialogue Attributes[15] are the semantic representation of their sentences and the discourse entities introduced by the sentences.

Rules to build the discourse structure: Given sentence(S1,Speaker) where S1 is the first sentence semantic representation, the update of the new sentence dialogue is:

s(basic(Speaker,S1)).

This fact gives rise to the update of the new Dialogue Structure according to the above rules:

1) ds(specify(O_{ds},D_s)) $\leftarrow$
ds(O_{ds})/past, s(Sp,D_s)/now,
bel(Sp,specify(O_{ds},D_s)/now.

2) ds(new(O_{ds},D_s)) $\leftarrow$
ds(O_{ds})/past, s(Sp,D_s)/now,
bel(Sp,new(ds(O_{ds},D_s))/now.

3) ds(repair(O_{ds},D_s) $\leftarrow$
ds(O_{ds})/past, s(Sp,D_s)/now,
bel(Sp,repair(O_{ds},D_s)/now.

These 3 rules encode that the new discourse structure is a structure that includes the semantics of the new sentence and that the systems is able to infer that at this point of the dialogue the speaker believes in that structure. The conditions for speaker beliefs will be presented later, but as it can be seen from discourse structure of dialogue 1 in figure 1: the system normally intends to specify the user previous sentence; and there are preferences for assuming that the user intends to specify the previous dialogue structure.

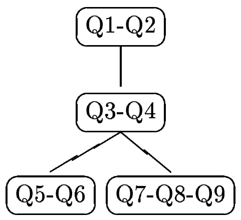

Figure 1: Dialogue Structure for Dialogue 1

The structure of dialogue 1 in figure 1 was built using these rules and it represents the dialogue structure after utterance Q9. The tree in the figure is the representation of following terms, where spf for specify.

ds(new(Q1))/t_0.	ds(spf(ds(D)/t_3,spf(Q5,Q6)))/t_5.
ds(spf(new(Q1),Q2))/t_1.	ds(spf(ds(D)/t_3,Q7))/t_6.
ds(spf(ds(D)/t_1,Q3))/t_2.	ds(spf(ds(D)/t_6,spf(Q7,Q8)))/t_7.
ds(spf(ds(D)/t_1,spf(Q3,Q4)))/t_3.	ds(spf(ds(D)/t_6,spf(Q7,spf(Q8,Q9))))/t_8.
ds(spf(ds(D)/t_3,Q5))/t_4.	

The system displays a representation of the dialogue similar to the one in figure 1, but without the system interventions in order to help the user to keep in mind the dialogue context. Moreover, our system allows the user to select a node in the tree for defining the context of his next query. This facility has showed to be very useful since our users use it very frequently. We are able to deal with it by translating the user act into a belief in a discourse structure for the next utterance.

4 Inference of user Intentions

In order to be collaborative our system needs to model user attitudes (intentions and beliefs). This task is also achieved through the use of logic programming framework rules and the dynamic LP semantics [12].

The system mental state is represented by an extended logic program that can be decomposed in several modules [14]:

- Description of the effects and the pre-conditions of the speech acts in terms of beliefs and intentions;
- Definition of behaviour rules that define how the attitudes are related and how they are transferred between the users and the system (cooperatively).

For instance, the rule which describes the effect of an inform and a request speech act from the point of view of the receptor (assuming cooperative agents) is:

bel(A,bel(B,P)) $\leftarrow$ inform(B,A,P)/bef.	bel(A,int(B,Action)) $\leftarrow$ request(B,A,Action)/bef.

In order to represent collaborative behaviour it is necessary to model how information is transferred from the different agents:

bel(A,P) $\leftarrow$ bel(A,bel(B,P))/now, (not bel(A,P))/bef.	int(A,Action) $\leftarrow$ bel(A,int(B,Action))/now, (not neg int(A,Action))/bef.

These two rules allow beliefs and intentions to be transferred between agents if they are not inconsistent with the previous mental state (neg stands for the explicit negation and not stands for the negation by omission).

After each event (for instance a user question) the agents' model needs to be updated with the description of the event that occurred. The dialogue system recognizes the speech act and it constructs the associated speech act (request or inform). The speech act will be used to update the logic program in order to obtain a new model. Using this new model it is possible to obtain the intentions of the system.

4.1 System reasoning steps

Any user act (utterance or other) will cause a system update that will give rise to following reasoning steps:

Update of the user act that may be an utterance or a menu choice or another multimodal act. Ex: if the act is Q1 from dialogue 1 it causes the update of: s(user,[x,y: documents(x), concept(y), y=pension, about(x,y)])/t_0

The update of the inference of the dialogue structure using the dialogue structure rules and the speech act of the user. Ex. for Q1 from dialogue 1: consider DRS1= [x,y:documents(x), concept(y), y=pension,about(x,y)]. The updates are: ds(new([],basic(user,DRS1)))/t_0 and request(u, s, inform(s, u, DRS1]))/t_0. The request is inferred from the action Q1 and the structure ds(new([], basic(user, DRS1)))/t_0

The update of system intentions: that are inferred from speech acts rules. Ex: for Q1 from dialogue 1 - int(s,inform(s,u, DRS1))/t_1. Due to the speech act rule for requests in the Interaction knowledge level.

The execution of system intentions: In order to execute the inform action we must obtain values for the free variables in the semantic representation.

A non collaborative version of the dialogue system will simple launch the query: "sino: search Y" at the Text knowledge level and the system output will be that of a normal information retrieval system.

Our collaborative version will use the Information Retrieval and the Domain level to predict the user goals. In the next section we present how this can be achieved.

5 Prediction of user Goals

For the prediction of user goals the system uses 3 representation levels: Domain Knowledge; Information Retrieval; and Text level.

The Domain level is used to obtain the concepts to be searched.

Ex:pension(X) will give the models: {pension, military}, {pension, civilian}, {pension}.

They are computed assuming that we have the domain rules:

pension(X) ← military(X), action(X,A), behind_duty(A).	pension(X) ← civilian(X), action(X,A), save_life(Y,A), life_at_risk(X,A), X≠Y.

This knowledge level is built using the Laws describing the requisites for some juridical condition. For instance the law describing the requisites to obtain a pension for relevant services can be encoded by the previous rules. These rules state that:

— A military may have a pension for relevant services if he has been the agent of an action, and that action was behind is duty.

— A civilian may have a pension for relevant services if he has been the agent of an action that saves another life putting his live at risk.

In order to allow the system to obtain the possible explanations of the user queries, we define attributes as abducible predicates. Using this approach it's possible to obtain the set of non-contradictory logic models that explain the user query.

These models are used to ask the user to supply the value of those predicates: request(s, u, inform(u, s,[document(Y),concept(Z), Z=military, about(Y,Z)])), request(s, u, inform(u, s,[document(Y),concept(Z),Z=civilian, about(Y,Z)])).

This is how the system generated table 2 in Q2.

The Text level is where the system launches the query with the concepts, for our example it is: Y PENSION. The result of this query is used in the information retrieval and to inform the user.

Information Retrieval Level is built with rules that can be obtained by processing the text documents looking for keywords that give rise to disjoint sets of documents.

Example of rules:
pension(X) ← pension_given(X).
pension(X) ← pension_refused(X).
false ← pension_refused(X), pension_given(X)

These rules state that a document with the concept pension either mentions the concept attributed or rejected. Table 1 of Q2 was generated using these rules.

The result of the Text level is used to compute new rules from the clustering of the set of documents selected by the user query. This is how table 3 of Q2 was generated.

Note that 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, 8] that clusters documents into topically-coherent groups. It is able to present descriptive textual summaries that are build with topical terms that characterize 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 application the topical terms that characterize the clusters are chosen from the expressions in our juridical thesaurus.

6 Conclusions and Future work

We have presented a framework for a dialogue information retrieval system that is able to collaborate with the users in order to refine their queries. Collaboration is achieved through the inference of the user attitudes, the dialogue structure, and the clustering of the retrieved documents. Our framework was implemented over a legal web information retrieval system and it has been used by the public since the beginning of the year 2000.

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 try to do it yet, but the main issues are:

— A robust natural language grammar enabling 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 modelling some domain knowledge. If the system does not have this domain rules it still can act 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.

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, 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.
5. G. Greenleaf, A. Mowbray, and G. King. Law on the net via austlii - 14 m hypertext links can't be right? In *In Information Online and On Disk'97 Conference, Sydney*, 1997.
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. S. McRoy and G. Hirst. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics*, 21(4), 1995.
12. L. M. Pereira and P. Quaresma. Modeling agent interaction in logic programming. In *Proc of the 11th Int. Conf. on Applications of Prolog*, Tokyo, Japan, 1998.
13. Martha Pollack. Plans as complex mental attitudes. In Philip Cohen, Jerry Morgan, and Martha Pollack, editors, *Intentions in Communications.* MIT Press Cambridge, 1990.
14. P. Quaresma and J. G. Lopes. Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. *Journal of Logic Programming*, 54, 1995.
15. I. P. Rodrigues and J. G. Lopes. Building the text temporal structure. In *Progress in Artificial Intelligence: 6th EPIA*. Springer-Verlag, 1993.
16. Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer.* Addison-Wesley, 1989. Reading, MA.