An Information Retrieval System

with Cooperative Behavior

Paulo Quaresma and Irene Pimenta Rodrigues
pq@di.uevora.pt   ipr@di.uevora.pt

Departamento de Informática
Universidade de Évora
7000 Évora
Portugal

CENTRIA/AI Center
Universidade Nova de Lisboa
2825 Monte da Caparica
Portugal

Abstract

In this paper we will present a system that is able to perform cooperative information retrieval actions over a text knowledge base. The knowledge base is composed by four levels: Interaction, Domain, Information Retrieval and Text. The interaction level is responsible for the dialogue management, including the inference of attitudes. The domain level is composed by rules encoding knowledge about the text domain. The information retrieval level includes knowledge about IR actions over sets of documents. The text level has knowledge about the words in each text.

Cooperation is achieved through two main strategies: 1) clustering the answer sets of documents accordingly with the domain and IR-level knowledge; 2) keeping the context of the interaction and inferring the user intentions.

1. Introduction

In this paper we present a cooperative information retrieval system in the law domain. The information retrieval system is composed by a text knowledge base built from the Portuguese Attorney General documents.

As an example, the following dialogue demonstrates the major features of our system (the example will be explained in more detail in the last section).

User - U1: Documents about pensions for relevant services?
System - U2: Pensions that were given or refused?
User - U3: Both.
System - U4: Pensions for militaries or civilians?
User - U5: Civilians
System - U6: There are 129 documents, listed in the right frame.
User - U7: where there was a drowning?
System - U8: Since there are 45 documents of the 129 above that match the concept, I've listed them in the usual frame.
User - U9: No, I mean to look for all documents that match the concept.
System - U10: There are 347 documents listed in the right frame.
User – U11: Documents about drugs?

System – U12: I assume that you want a new query about drugs since there are no documents with the concept drugs and pensions for relevant services for civilians. Do you want to see documents about the use of drugs while driving?

User: - U13 yes.

...

In order to handle dialogues similar with this one, our system needs to be able to keep the context of the interaction and to cooperatively supply suggestions for further refinement of the user queries. The refinement process is based on the domain knowledge and the ability to compute clusters of documents associating a keyword (from a juridical thesaurus with 6000 expressions) to each cluster.

In order to perform a cooperative interaction with the user, the system should be able:

- To infer what are the user intentions with the queries. For instance, when a user asks for documents with a particular keyword, he may be interested in documents that do not have that exact keyword and he may not be 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 user query in order to help him in the refinement of his query.

As a consequence our system needs:

- To record the user interactions with the system. User interactions will provide the context of sentences (questions and answers), allowing the system to solve some discourse phenomena such as anaphoras and ellipses.
- To obtain new partitions (clusters) of the set of documents that the user selected with his query(ies). The clustering process should be based on the text knowledge representation.

In the next section we will describe the text knowledge base. Then, in section 3 and section 4 the interaction structure and the inference of attitudes will be described. In section 5, the clustering process will also be described. In section 6, an example of a cooperative session will be presented. Finally, in section 7, conclusions and future work will be presented.

2. Knowledge Base

The knowledge base is composed by four levels: Interaction, Domain, Information Retrieval and Text.

1. 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;
2. The domain level includes knowledge about the text domain and it has rules encoding that knowledge. For instance, in the legal domain 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;

3. 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;
4. The Text Level has knowledge about the words and sequence of words that are in each text of the knowledge base. This level is based on SINO, a text search engine with inverted files from the AustLII Institute [Greenleaf et al. 1997] that was extended to the Portuguese language. The extended SINO is able to access a 900,000 Portuguese lexicon and it is able to handle morphological information (verbal forms, plurals, etc.)

These four levels of knowledge are integrated via a dynamic logic-programming module, which is responsible for the management of the interaction with the users.

Dynamic logic programming [Alferes et al. 1998] defines how a logic program can be updated by other logic programs. In our approach, each event is represented by a logic program (composed only by facts), which is used to update the previous program and to obtain a new one. In fact, events are represented by an update situation and there is no need to explicitly represent time points. Inertia rules are also guaranteed by the dynamic logic programming semantics.

3. Interaction structure

The system builds the interaction 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 interaction and to build a new branch from there.

The Interaction Structure (IS) is made of segments that group sets of interactions. At present we are able to deal with 3 different kinds of segments:

- Basic --- has 2 arguments: Speaker;Action Representation
- New --- has 2 arguments: Interaction Structure; Interaction Structure. The new IS inherits its attributes from the second argument. Ex: New([],basic(User,Q1))
- Specify --- has 2 arguments: Interaction Structure; Interaction Structure. The new IS inherits its attributes from both interaction structures. Ex: Specify(Basic(User,Q1), Basic(System, Q2))

3.1. Rules to build the interaction structure

Given an action A1 from an agent A, the update of the new action is:

action(basic(A,A1)).

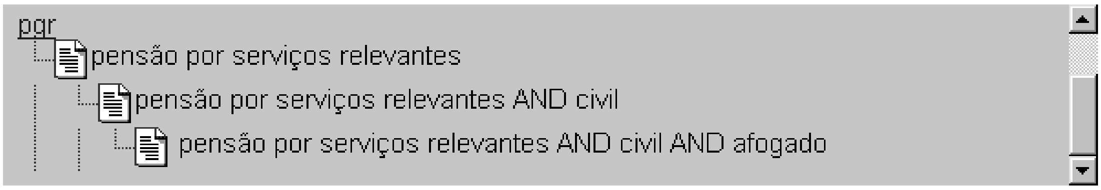
This fact gives rise to the update of the new Interaction Structure according to the above rules:

```
is(specify(Ois,Is)) <- is(Ois)/past, action(Is)/now,
                        bel(system,specify(Ois,Is)/now.

is(new(Ois,Is)) <- is(Ois)/past, action(Is)/now,
                        bel(system,new(Ois,Is))/now.
```

These two rules encode that the new interaction structure is a structure that includes the semantics of the new action and that the system is able to infer that at this point of the interaction is believable.
The conditions for this system belief can be defined in many different ways, but our system normally beliefs that users intend to specify previous actions.

```
bel(system, specify(Ois,Is)) <- not neg bel(system, specify(Ois,Is)).
bel(system, new(Ois,Is)) <- bel(system, incompatible(Ois,Is)).

neg bel(system, specify(Ois,Is)) <- bel(system, new(Ois,Is)).
neg bel(system, new(Ois,Is)) <- bel(system, specify(Ois,Is)).
```

Where "not" means default negation and "neg" means explicit negation.



As it is shown, the system displays a graphic representation of the interaction in order to help the user to keep in mind the interaction context. Moreover, it allows the user to select a node in the tree for defining the context of his next query. This feature has shown to be very useful since our users use it very frequently.

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 [Pereira and Quaresma 1998].

The system mental state is represented by an extended logic program that can be decomposed in several modules (see [Quaresma and Lopes 1995] for a complete description of these modules):

- 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)) <- inform(B,A,P)/before.
bel(A,int(B,Action)) <- request(B,A,Action)/before.

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

bel(A,P) <- bel(A,bel(B,P))/now, (not bel(A,P))/before.
int(A,Action) <- bel(A,int(B,Action))/now, (not neg int(A,Action))/before.

These two rules allow beliefs and intentions to be transferred between agents if they are not inconsistent with the previous mental state.

After each event (for instance a user question) the agents' model (logic program) needs to be updated with the description of the event that occurred. The 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:

1. Update of the user act;
2. Update of the new interaction structure using the interaction structure rules and the updated act;
3. Update of system intentions that were inferred from the effects of the action rules;
4. Execution of the system intended actions.

4.2. Cooperative inference of user Goals

In order to infer the user goals the system uses two representation levels: Domain Knowledge and Information Retrieval knowledge.

The Domain level

This level is used to obtain the domain models that are consistent with the user query.

Example:

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 from the laws used in the texts. 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 someone life and he has his live at risk.

These models are used to ask the user to refine his query.

Information Retrieval Level

This level of knowledge is built with rules that can be obtained by processing the text documents looking for keywords that give rise to disjoint sets of documents. By now we obtain these rules using a thesaurus with keywords for text juridical classification.

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.

As it was explained in the previous sub-section, the system computes the models that are consistent with the user query and with the interaction structure. Then, if there are more than one possible model, the system asks the user to refine his query and to refine it.

5. Clustering

A service that an information retrieval system always provide is the organisation of retrieval results. SINO can also rank the output according to estimated relevance values. However, if there are large document groups with similar rankings, it is a good option to build clusters and to label the groups with some relevant keywords.

Clustering helps users by showing them some kind of pattern in the distribution and it allows the user to include or exclude sets of documents from further searches.

We use the already existing juridical classification of the Portuguese Attorney General documents to characterize each cluster and to inform the user on possible further choices for refining his query.
Our clustering method is based on the Scatter/Gather browsing paradigm that clusters documents into topically-coherent groups. It is able to present descriptive textual summaries that are build 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.

6. Example

The following dialogue is an example of a user session in our system, which demonstrates its major features.

User - U1: Documents about pensions for relevant services?
System - U2: Pensions that were given or refused?
User - U3: Both.
System - U4: Pensions for militaries or civilians?
User - U5: Civilians
System - U6: There are 129 documents, listed in the right frame.
User - U7: where there was a drowning?
System - U8: Since there are 45 documents of the 129 above that match the concept, I've listed them in the usual frame.
User - U9: No, I mean to look for all documents that match the concept.
System - U10: There are 347 documents listed in the right frame.
User – U11: Documents about drugs?
System – U12: I assume that you want a new query about drugs since there are no documents with the concept drugs and pensions for relevant services for civilians. Do you want to see documents about the use of drugs while driving?
User: - U13 yes.
...

Given a user utterance, such as U1, the system is able to cooperatively interact with user in order to refine its query.
The system reply to U1 will be U2. This reply is achieved by recognizing that this query can be refined since the texts that mention pensions can be divided into two disjoint sets, one where pensions were given and another one were pension were refused. This kind of knowledge is encoded in what we have called the Information Retrieval level.
After the user answer (that could be: given, rejected or both), by using knowledge of the Domain level the system will generate question U4. This is achieved by knowing that pensions by relevant service have different conditions when there is a military or a

civilian. This is juridical knowledge independent of the texts present in the text base and represented in the domain level.

As we described in previous sections, the system is also able to decide if the user intends to continue its previous query (its utterance is to be interpreted in the context of the previous dialogue) or to open a new query (a new interrogation context).

If, after U1 the user asks U7, the system will be able to decide that the user intends to look for text where there are a pension and a drowning. But if the user utters U11 instead of U7 the system will conclude that the user intends to open a new interrogation context.

This is achieved by using the Textual level that encodes knowledge about the texts words and expressions (concepts). Using our retrieval information system SINO, it is possible to see that are some texts where the concepts *pension and drowning* appears but no texts where the concepts *pension and drugs* appears. This is what the user expects the system behave in most cases. When this is not the case the user may clarify its query in order oblige the system to behave differently. For instance after U7 the system will reply U8 and the user may reply U9.

U9 will be understood by the system as a user clarification and it will forget the semantic content of sentences U1-U8 by opening a new context with U9. In order to interpret the sentence U9 in particular to solve the nominal anaphora *the concept*, the dialogue structure of sentences U1-U8 will be used.

7. Conclusions and Future Work

The Information Retrieval system presented in this paper is implemented and it has a public access from the WWWeb.

A preliminar evaluation was done by taking into account the system logs and user comments. By analyzing the system logs we obtained that:

- 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 booleanconnections).
- 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.

Regarding the portability of our IR system into other domains, the main issues are:

- A robust natural language grammar enabling to obtain the speach act associated to a user multimodal act. (it may involve to add some vocabulary and some knowledge representation rules, mainly a domain thesaurus).
- A knowledge base modeling some domain knowledge.
- The computation on-the-fly of document clusters with a topical expression associated with each. This will be our main source of knowledge to compute the system sugestions for further refinement.

References

Allen, J. & Kautz, H & Pelavin, R. & Tenenberg, J. 1991. *Reasoning about Plans*. Morgan Kaufman Publishers, Inc..

Alferes, J. & Pereira, L. 1996. Reasoning with Logic Programming. *Lecture Notes in Artificial Intelligence*, 1111. Springer.

Alferes, J. & Leite, J. & Pereira, L. & Przymusinska, H. & Przymuzinski, T. 1998. Dynamic Logic Programming. *Proceedings of KR'98- Knowledge Representaion*.

Cohen, P. & Levesque, H. 1990. Intention is choice with commitment. *Artificial Intelligence*, 42(3).

Carberry, S. 1988. Modelling the user's plans and goals. *Computational Linguistics*, 14(3):23--37.

Greenleaf, G. & Mowbray, A. & King, G. 1997. Law on the net via AustLII - 14 M hypertext links can't be right? *Proceedings of Information Online and On Disk'97*.

Grosz, B. & Sidner, C. 1986. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175--204.

Kamp, H. & Reyle, U. 1993. *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: D. Reidel.

Litman, D. & Allen, J. 1987. A plan recognition model for subdialogues in conversations. *Cognitive Science*, (11):163--200.

Lascarides, A. & Asher, N. 1991. Discourse relations and defeasible knowledge. *In Proceedings of the 29th Annual Meeting of ACL*, 55--62.

Pinto, J. & Reiter, R. 1993. Temporal reasoning in logic programming: A case for the situation calculus. *Proceedings of the 10th ICLP*. MIT Press.

Pereira, L. & Quaresma, P. 1998. Modeling Agent Interaction in Logic Programming. *In Proceedings of the 11th International Conference on Applications of Prolog*. Tokyo, Japan.

Quaresma, P. & Lopes, J. 1995. Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. *Journal of Logic Programming*, (54).

Rodrigues, I. & Lopes, J. 1992. Discourse temporal structure. *In Proceedings of the COLING'92*.

Rodrigues I. & Lopes, J. 1993. Building the text temporal structure. *In Progress in Artificial Intelligence: 6th Portuguese Conference on AI*. Springer-Verlag.

Rodrigues, I. & Lopes, J. 1997. AI5, An Interval Algebra for the temporal relations conveyed by a text. *In Mathematical Linguistics II*, Eds Carlos Martin-Vide, John Benjamins.

Song, F. 1991. A Processing Model for Temporal Analysis and its Aplication to Plan Recognition. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada.