A Logic Programming Framework for Cooperative Information Retrieval

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

May 30, 1999

Abstract

We propose a framework for a cooperative information retrieval system based on dynamic logic programming. In order to be cooperative our system keeps the context of the interaction and tries to infer the user intentions. Each event is represented by logic programming facts which are used to dynamically update the previous user model. User intentions are inferred from this new model and are the basis of the interaction between the system and the legal texts knowledge base. As texts knowledge base we are using the documents from the Portuguese Attorney General (Procuradoria Geral da República). In this paper we will show some examples of the obtained cooperative behaviour.

1 Introduction

In this paper we propose a framework for a cooperative legal information retrieval system based on dynamic logic programming [1]. In our framework the user model is represented by a logic program and each event is described by logic programming facts which are used to dynamically update the previous user model. The result of this update action is a new logic program which models the interaction status. Collaboration is achieved through the inference of user intentions using rules describing the legal domain. The main idea is to help the user to access the legal text knowledge base. For instance if the user asks the system to be informed about documents with a specific property P and that property can be a consequence of other properties A and B, the system will answer the user question but it will add information about documents where properties A and B hold.

As legal texts knowledge base we are using documents from the Portuguese Attorney General (Procuradoria Geral da República) and we are working with examples using the legislation that defines when a person has a right for a pension for exceptional services. Our information retrieval system is based on SINO, a search engine for legal text databases [4].

So, our cooperative IR framework is composed by the following modules:

- The SINO search engine indexing all words in the documents;
- A juridical terms taxonomy;
- A logic programming module to represent and to infer about user attitudes;

- A dynamic logic programming module which is responsible to update the user model with new events;
- A logic programming module to represent and to infer about legal knowledge;
- A top-level logic programming module which is the responsible for the coordination between the other modules.

After a user event, the following actions are done (under the responsibility of the top-level LP module):

1. Update the user model (dynamic logic programming module);
2. Infer the user intentions (attitudes LP module);
3. Infer other possible cooperative queries (legal knowledge LP module);
4. Interact with the IR system (SINO and juridical terms taxonomy);
5. Answer the user.

In the next section we will describe the IR framework (SINO and the juridical terms taxonomy). In section 3 the user attitudes representation module is described. In section 4 we will describe the dynamic logic program module and in section 5 we will show how legal knowledge can be represented using logic programs. Finally, in section 6 an example of a cooperative IR session is presented. Conclusions and future work are discussed in section 7.

2 Information Retrieval System

The information retrieval system is based on SINO [4] from the AustLII Institute. We have changed SINO in order to be adapted to the Portuguese Language. Namely, the new system uses the Portuguese lexicon (more than 900,000 words) in order to handle morphological errors and to obtain the base queried word. For instance, if the user asks to be informed about documents where a specific word appear, the systems also searches for documents containing derived words (plurals for nouns, verbal forms for verbs, ...).

As a top layer over the basic IR system we are using a juridical terms thesaurus. This thesaurus is a result from another 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 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 [9]).

3 User attitudes

In order to be cooperative our system needs to model user attitudes (intentions and beliefs) [3]. This task is achieved through the use of a logic programming framework that allows non-monotonic reasoning (well-founded semantics of extended logic programs with explicit negation, WFSX, from the work of Pereira et al. [2]). In this framework the system mental state is represented by an extended logic program that can be decomposed in several modules (see [8, 7, 6] 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 behavior rules that define how the attitudes are related and how they are transferred between the users and the system (cooperatively).

At present we are only considering the following speech acts:

- request
- inform

The rule which describes the effect of an inform speech act from the point of view of the receptor (assuming cooperative agents) is:

$$bel(A, bel(B, P)) \leftarrow inform(B, A, P).$$

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

$$bel(A, int(B, Action)) \leftarrow request(B, A, Action).$$

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

$$\begin{aligned} bel(A, P) &\leftarrow bel(A, bel(B, P)), \\ &\quad\; not\ \neg bel(A, P). \end{aligned}$$

$$\begin{aligned} int(A, Action) &\leftarrow bel(A, int(B, Action)), \\ &\quad\; not\ \neg int(A, Action). \end{aligned}$$

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

The system attitudes are the set of beliefs and intentions that are supported by the logic program P:

$$\begin{aligned} \{bel(system, A) &: P \models_{WFSX} bel(system, A) \\ \{int(system, A) &: P \models_{WFSX} int(system, A) \end{aligned}$$

On the other hand, the system's beliefs about the user attitudes are:

$$\begin{aligned} \{bel(system, bel(user, A)) &: P \models_{WFSX} bel(system, bel(user, A)) \\ \{bel(system, int(user, A)) &: P \models_{WFSX} bel(system, int(user, A)) \end{aligned}$$

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.

For the moment we are only considering as possible user intentions:

- to be informed on the number of legal texts that describe a situation (How many ....)
- to be informed on the names of legal texts that describe a situation (Which ....)

The interaction system recognizes the speech act and it constructs the associated speech act (request or inform). As it will be described in the next section, 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 system intentions:

$$\{int(system, A) : P \models_{WFSX} int(system,)$$

and to plan the system actions. In order to execute the intended actions, the system may need to interact with the text knowledge base (see section 6).

4 Dynamic Logic Programming

As it was pointed out in the previous section, after each event it is necessary to update the logic program which models the system mental state with the new logic program facts. This update process is done using dynamic logic program updates which allows us to update a logic program P with another logic program U ([1]). Using this approach a new event in the interaction process forces and update of the logic program creating a new state of knowledge (a new mental state).

In our framework the system initial mental state is defined by the logic program which includes the rules for the speech acts and for the cooperative behavior (see previous section). The action rules have to be changed in order to deal with time (update states):

$$\begin{aligned} bel(A, bel(B, P)) &\leftarrow inform(B, A, P)/before. \\ bel(A, int(B, Action)) &\leftarrow request(B, A, Action)/before. \end{aligned}$$

$P/before$ means that the predicate P should be verified in the previous update state. So, these action rules mean that the effects of an action are valid on the states after the action.

For instance, suppose the user asks for documents about accidents:

$$request(user, system, inform(system, user, X : documents(accidents))).$$

This event originates a program update and a new state ($S1$) where the fact is true. In order to plan its actions the system will generate a new update with an empty logic program and a new state will be obtained ($S2$). In $S2$ the following properties are valid:

$$\begin{aligned} bel(system, int(user, inform(system, user, X &: documents(accidents)))). \\ int(system, inform(system, user, X &: documents(accidents))) \end{aligned}$$

using the *request* action and the cooperative rule.

So, the system should try to perform the *inform* action. In order to do that it will be necessary to interact with the text knowledge base through the SINO module

$$X \leftarrow search\ accidents;$$

and to answer the user terminating the system intention:

$$\begin{aligned} &inform(system, user, X : documents(accidents)). \\ &\neg int(system, inform(system, user, X : documents(accidents))) \end{aligned}$$

The system answer will create a new update and a new mental state ($S3$).

5 Legal Knowledge Representation

In order to more cooperative our system needs the capability to model legal knowledge. We are using the extended logic programming framework described in the previous sections to represent some aspects of this knowledge. For the moment we have chosen only some very specific domains, namely, the legislation that defines when a person has a right for a pension for exceptional services.

Our main goal is to handle situations like this one:

- The user asks for documents were O is valid;
- The system knows (using the legal knowledge) that O can be valid in two different situations:

$$\begin{array}{rcl} O & \leftarrow & A \\ O & \leftarrow & B \end{array}$$

- The system gives the answer for documents where O is valid but also for documents where A and B are valid:
  - X documents where O
  - Y documents where O and A
  - Z documents where O and B

Using this approach the system is trying to be cooperative dividing the answer in clusters and predicting possible future user questions.

This system behavior can be modeled through the use of abduction in the extended logic programming framework. The goal is to abduce the predicates $\triangle$ needed to support the observation (O):

$$P \cup \triangle \models_{WFSX} O$$

where P is the logic program and each predicate in $\triangle$ belongs to the abducible set of predicates.

So, the cooperative behavior is achieved through the following steps:

- Define the legal rules:

$$\begin{array}{rcl} P_1 & \leftarrow & A_1, ..., A_n. \\ & & ... \end{array}$$

- Define the set of abducibles (set of antecedents — possible causes for the consequences):

$$\{A_1, ..., A_n, ...\}$$

- For each intention:

$$int(system, inform(system, user, X : documents(Y)))$$

  create an hypothetical integrity constraint:

$$Y \leftarrow$$

  and obtain the set of abduced models (*Abd*) which satisfy the constraints.

- For each model M_i in the abduced set *Abd* create a new intention:

$$int(system, inform(system, user, X : documents(Y \; and \; M_i)))$$

- Answer every intention!

6 Example

In this section we will show an example over the legislation that defines when a person has a right for a pension for exceptional services. However, due to its extension and complexity, we will make many simplifications over the legislation:

$$\begin{aligned}
pension &\leftarrow exceptional_war_action. \\
pension &\leftarrow exceptional_action. \\
exceptional_war_action &\leftarrow war_action, \\
&\quad\; beyond_standard_military_actions. \\
beyond_standard_military_actions &\leftarrow defense_other_lives_above_his_own. \\
exceptional_action &\leftarrow benefits_country, \\
&\quad\; beyond_duty_of_functions.
\end{aligned}$$

These rules mean that a person has the right for a pension for exceptional services if has done an exceptional war actions or an exceptional action. A war action is considered exceptional is it is a war actions and it is beyond the standard military actions. An action is considered exceptional if it benefits the country and it is beyond the duty of functions.

Analyzing these rules we obtain the set of abducibles (predicates that can not be derived from the rules):

$$\begin{aligned}
&\{war_action, \\
&defense_other_lives_above_his_own, \\
&benefits_country, \\
&beyond_duty_of_functions\}
\end{aligned}$$

Suppose the user asks the following question:

$$request(user, system, inform(system, user, X : documents(pension))).$$

Using the rules presented in the section 3 and the update process of section 4 we will obtain a new state where the following property is valid:

$$int(system, inform(system, user, X : documents(pension)))$$

As the methodology described in the previous section proposes, we add an hypothetical integrity constraint to our model:

$$pension \leftarrow$$

And we calculate the abduced models:

$$\{war\_action, defense\_other\_lives\_above\_his\_own\}$$
$$\{benefits\_country, beyond\_duty\_of\_functions\}$$

Finally it is created a new state through the update of the previous program with the following facts:

$$\begin{aligned}
int(system, inform(system, user, X &: docs(pension\ and\ war_action\ and \\
&\quad defense_other_lives_above_his_own))). \\
int(system, inform(system, user, X &: docs(pension\ and\ benefits_country\ and \\
&\quad beyond_duty_of_functions)))
\end{aligned}$$

In order to execute the intended actions the system will create the following SINO queries:

$search\ pension$;

$search\ pension\ and\ war\_action\ and\ defense\_other\_lives\_above\_his\_own$;

$search\ pension\ and\ benefits\_country\ and\ beyond\_duty\_of\_functions$;

However, there are some problems with these SINO queries: for instance how should the concept "benefits country" be searched? For the moment we use the juridical thesaurus and the synonymous dictionary to expand each word and we search for the set of words. This a naive approach and, as future work, we will need to analyze parts of the documents and to transform them into a logical representation (DRS [10, 11, 5]) which can be used by our logic programming framework.

7 Conclusions

The proposed framework has the following main characteristics:

- It is able to represent the system "mental state": attitudes, actions description, events;
- It uses dynamic logic programming in order to update the system mental state with new events;
- It uses logic programming to represents legal knowledge;
- It uses abduction in order to infer possible causes for properties;
- It gives cooperative answers to the user dividing the answers into clusters and trying to predict possible future intentions;
- It uses a IR system (SINO) specialized in the Portuguese Language to search the textual knowledge bases.

However, much work has still to be done:

- In order to process the documents (or at least parts of them) we need to improve the natural language processing tools (taggers, corpora, syntactic and semantic analyzers)
- The domain knowledge must be represented in a more extended way.

References

[1] José Júlio Alferes, Jo ao Alexandre Leite, Luı́s Moniz Pereira, H. Przymusinska, and T. C. Przymusinski. Dynamic logic programming. In S. Shapiro A. Cohn, L. Schubert, editor, *Proceedings of KR'98 - Knowledge Representation*, pages 98–109, 1998.

[2] Jose Julio Alferes and Lus Moniz Pereira. *Reasoning with Logic Programming*, volume 1111 of *Lecture Notes in Artificial Intelligence*. Springer, 1996.

[3] P. Cohen, J. Morgan, and M. Pollack. *Intentions in Communication*. MIT Press, Cambridge, MA, 1990.

[4] 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.

[5] Hans Kamp and Uwe Reyle. *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: D. Reidel, 1993.

[6] Gabriel Lopes, Paulo Quaresma, and Irene Pimenta Rodrigues. Question/answer dialogues for interfacing a database with supreme court decisions. In *Proceedings of ICAIL'97 - International Conference on AI and Law - (poster)*, 1997.

[7] Paulo Quaresma. *Inference of Attitudes in Dialogues.* PhD thesis, Universidade Nova de Lisboa, 1997.

[8] Paulo Quaresma and José Gabriel Lopes. Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. *Journal of Logic Programming*, 54, 1995.

[9] Paulo Quaresma and Irene Pimenta Rodrigues. Keeping context in web interfaces to legal text databases. In *Proceedings of 2nd French-American Conference on AI and Law*, 1998.

[10] Irene Pimenta Rodrigues and José Gabriel Lopes. Discourse temporal structure. In *Proceedings of the COLING'92*, 1992.

[11] Irene Pimenta Rodrigues and José Gabriel Lopes. Building the text temporal structure. In *Progress in Artificial Intelligence: 6th Portuguese Conference on AI.* Springer-Verlag, 1993.