PGR Project: The Portuguese Attorney General Decisions on the Web

Paulo Quaresma and Irene Pimenta Rodrigues
and Gabriel Lopes
email:(pq,ipr,gpl)@di.fct.unl.pt
AI Center, Departamento de Informática
Universidade Nova de Lisboa
Quinta da Torre, 2825 Monte da Caparica,
Portugal

Teresa Almeida and Elsa Garcia and Ana Lima
email: (talmeida,egarcia,analima)@pgr.pt
Procuradoria Geral da República
Rua da Escola Politécnica, 140
1294 Lisboa Codex,
Portugal

Abstract

In this paper we describe a web based information retrieval system Portuguese for a database with the Portuguese Attorney General documents. The text search engine is specialized for the Portuguese language, and it takes into account lexical, syntactic, and (some) semantic information: equivalent, related, more specific or more general relations between juridical expressions. These relations between juridical terms are used to manipulate the user queries and the system answers. Basically we try to expand the queries in order to take into account related and specific concepts and, then, we try to collapse the answers in order to obtain more general results.

The system also tries to keep an interaction context joining, whenever is possible, the new user query with the previous one, allowing the refinement of queries in a very friendly way.

Moreover, the system allows the representation of legal knowledge using a logic programming language and it allows us to check the rules against specific documents.

1. Introduction

In this paper we present a system that has an intelligent web interface for a database with legal texts, the Portuguese Attorney General texts. The main purpose of the interface is to allow the search and the view of the database documents. The user poses a query and it should obtain as an answer the documents set in the database that satisfy the query, as well as some proposals for further refinement of the query.

The database has a set (aprox. 7000) of documents that are structured into sections that include: the conclusions, integral text, administrative information, and the juridical classification.

This set of documents was processed using SINO, a search engine for legal text databases (Greenleaf, Mowbray and King 1997). We have changed SINO in order to handle some features of the Portuguese language, namely stop words, plurals, verbs, and synonyms.

In order to deal with these features of the Portuguese, we use a lexical dictionary that was built by our team in FCT/UNL that as been augmented with the vocabulary of the Attorney General texts. In order to obtain a correct treatment of the verbs and nouns, we also have to use a Portuguese tagging that marks the lexical category of the words in the context (if it is a noun or a verb). This tagging and dictionary are used in the process of indexing the texts with Sino and is used in processing user query. This way we are able to avoid the reference to unintended documents due to lexical ambiguity (nouns that are written like some verb forms).

Our system is a hybrid system in the sense that it uses different sources of knowledge in order to build the user answer and the proposals for query refinement. Namely it uses our juridical terms

thesaurus and a knowledge base with juridical rules. The Attorney General documents have a juridical classification section that contains a set of pre-defined juridical terms, descriptors, that result from the juridical analysis made by the PGR staff. Our PGR partners have been studied those juridical terms, descriptors, in order to build a thesaurus with the relations: *equivalent*, *specifies*, *generalizes* and *is related with*. Given the documents classification, the thesaurus and a set of documents it is possible to group sets of documents that have the same descriptor or a related descriptor (equivalent, generalizes or related with). This is one way of proposing possible query refinement to the user. This feature has particular interest to the users that are not aware of the all set of descriptors and their use, law students and layers not working at PGR.

Another use of our juridical terms thesaurus is in the expansion of queries given a word or expression that is in the thesaurus, that word is expanded into a set of words or expressions using the relations in the thesaurus.

Our thesaurus is also used to give an overview of all documents in the database with the structure given by the descriptors that they refer. We also provide another structure of the documents in the database using the information where they refer to other database documents.

These views of the database documents are very useful to users without law knowledge.

By now, our web interface supports queries in SQL like Boolean expressions, allowing the user to state where does he want to look for the occurrence of a word or expressions: in the all text or in a particular section. We try to maintain the interrogation context in both kind of queries and treat the user query/answer sequence as a dialogue between two intelligent agents.

2. The Juridical Terms Thesaurus

As it is common to other documents classification in libraries, the PGR documents are the subject of a juridical analysis by PGR staff. Each document is classified with one or more descriptors from a set of descriptors that has been augmented over the years, by now there are 6000 descriptors. When this project has started there were no efforts in order to group or to relate them. Since we build the project this has become one of our main tasks since the juridical knowledge contained in those expressions (descriptors) can be explored in many ways.

In this section we first present the organization of the thesaurus and then we present the ways it is being used in our system.

The relations in the thesaurus

For each descriptor we try to list the descriptors that are in one of the following four relations:

- Is equivalent to
  Ex: **law** is equivalent to **norm**
- Is generalized by
  Ex: **prime minister** is generalized by **minister**
- Is specified by traffic
  Ex: **accident** is specified by **traffic accident**
  disaster
- Is related with
  Ex: **desertion** is related with **traffic accident**
  army

The properties of these relations are:

- ***Equivalent***
  1. A **is equivalent to** B => B **is equivalent to** A
  2. A **is equivalent to** B**,** B **is equivalent to** C => A **is equivalent to** C
- ***Generalized***

1. A **is specified by** B => B **is generalized by** A
2. A **is generalized by** B, B **is equivalent to** C => A **is generalized by** C
3. * A **is generalized by** B, B **is related with** C => A **is generalized by** C

- ***Specified***
  1. A **is generalized by** B => B **is specified by** A
  2. A **is specified by** B, B **is equivalent to** C => A **is specified by** C
  3. * A **is specified by** B, B **is related with** C => A **is specified by** C
- ***Related***
  1. A **is related with** B => B **is related with** A

Given an initial list of descriptors, their relations and the relations properties it is possible to compute the closure and to obtain a new list that can be checked by PGR specialists to see the consequences of their initial definitions. This has been our procedure in the task of building the thesaurus.

Uses of the thesaurus

The juridical knowledge contain in the thesaurus by now is being used in 3 different modules:

- Processing queries
- Computing proposals for query refinement
- Overview of the documents

These 3 modules will now be described.

Processing queries

The knowledge in the thesaurus is used to expand the user queries. Whenever a query that specifies the value of a descriptor is made, we expand it with all the values that are: equivalent or more specific or related, with the initial descriptor value.

Suppose a user wants to be informed on legal texts about "accidents". It may pose different queries such as the ones below:

- Accident
- Integral_Text(accident) or Conclusions(accident)
- Descriptor(accident)

The system expands the descriptor in each query using the thesaurus adding the following descriptors:

- all the descriptors equivalent to accident
- all the descriptors that are in the transitive closure of the relation:

  ‘accident’ *is specified by* …
- all the descriptors related with accident

In this case, it will substitute the word accident for "accident OR traffic accident OR desertion

OR disaster…"

A result of this procedure the users may obtain documents where the word accident is not present in any section but the document will potentially be about an accident. The cases where we obtain worse results (documents that are not about accidents) are those that result from the expansion with the relation "is related with". Even with the addition of irrelevant documents our users still find this procedure very useful because when the systems proposes query refinement it is easy to eliminate those irrelevant documents.

Computing proposals for query refinement

Whenever we have a large set of selected documents, we compute proposals for query refinement. These proposals are presented as a list of descriptor-number of documents. The user may select one or more items from the list and launch another query whose meaning is initial query and the ands of the descriptors in each selected item.

Since the above list of proposals may be very large, we may need to collapse sets of 'descriptor-number of documents'. The collapse of the initial set of descriptors will be done with following procedure:

Given D1, the initial set of pairs 'descriptor-{set of selected documents}'we build a new set D2 by:

- including in D2 a pair $desc_n$-S_n

  if there are two pairs ' $desc_i$-S_i, $desc_j$-S_j' in D1 such that the descriptors $desc_i$ and $desc_j$ verify the relations: {$desc_i$ *is specified by* $desc_n$ and $desc_j$ *is specified by* $desc_n$}**and** $S_n=S_i+S_j$.

  Or if there are two pairs $desc_n$-S_{ni}, $desc_j$-S_j' in D1 such that the descriptors $desc_n$ and $desc_j$ verify the relation: $desc_j$ *is specified by* $desc_n$ **and** $S_n=S_{ni}+S_j$.

  Or if the pair $desc_n$-S_n is in D1 and there is no descriptor in D1 or D2 that can be related with it by the relations: *is specified by* or *is generalized by*.

- Changing the pair $desc_n$-S_m in D2 into the pair $desc_n$-S_n

  if there is a pair'$desc_i$-S_i in D1 such that the descriptors $desc_i$ and $desc_n$ verify the relations: {$desc_i$ *is specified by* $desc_n$}**and** $S_n=S_i+S_m$.

This procedure can be applied into a set of pairs descriptor-{set of selected documents}'as many times as we need until we obtain the desired cardinal for the set of pairs.

With this procedure the proposals for query refinement will correspond to groups of documents that are juridical related and classified. This feature is very useful to our users since it supplies new keywords with strong juridical meaning that the users may not be aware or may not have it present when they build the query.

Example

Suppose a user wants to be informed about legal texts about "accidents":

- accidents?

The system expands the query using the thesaurus and it searches for all the related and more

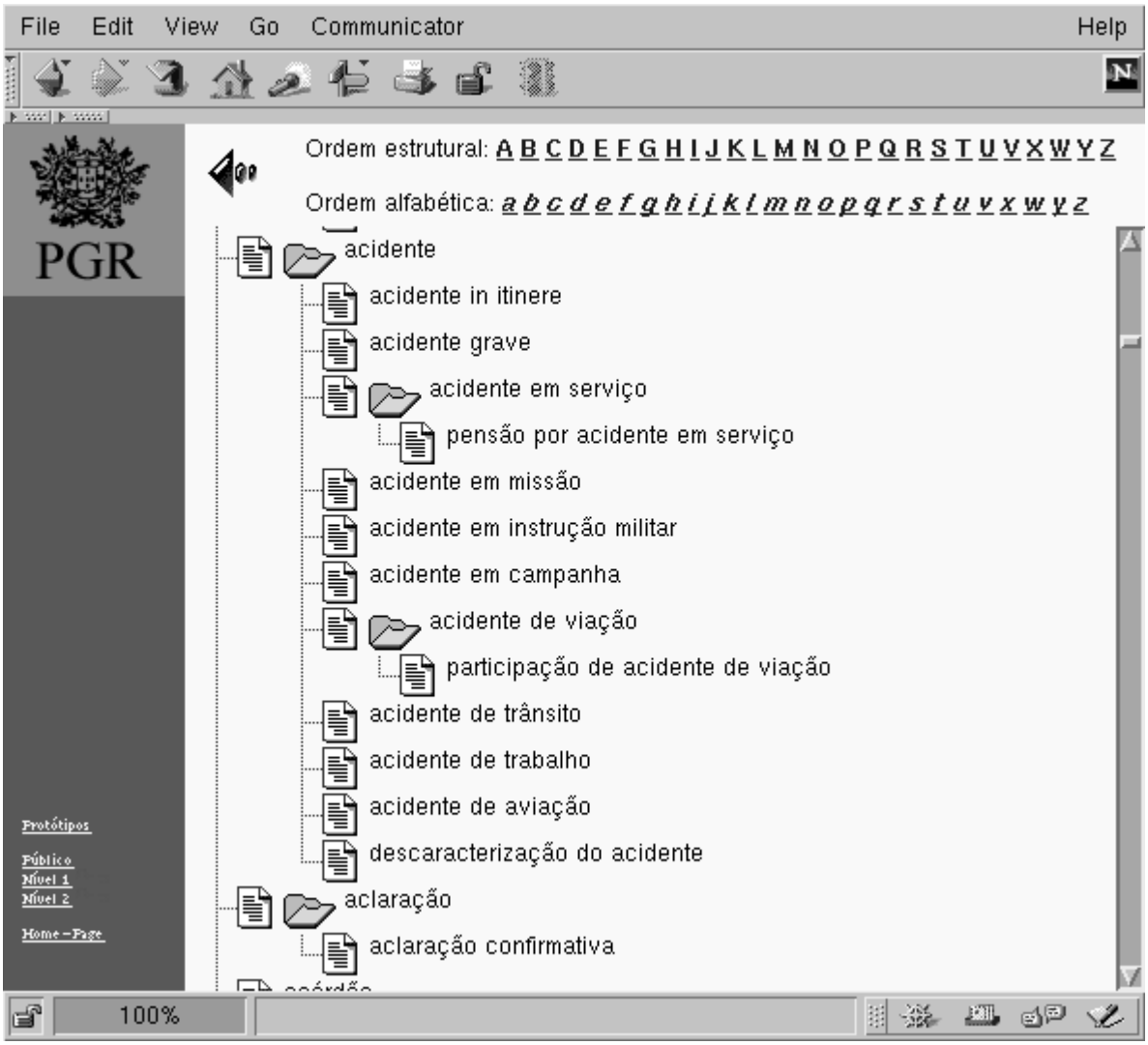
specific values. For instance, it will search for "accident OR traffic accident OR desertion OR ….."

Then, the answer is used to collapse the set of texts into classes of answers grouped by the thesaurus terms:

- X documents about accidents;
- Y documents about desertion;
- Z documents about civil damage
- W documents about criminal damage
- etc ...

Overview of the documents

The thesaurus structure will allow us to display the set of all documents in a structured and meaningful way into our users.



3. Query Context

Before processing the user query our system tries to obtain the context of the interaction. This task is done in the following steps:

1. Join the query with the previous one (if exists). The joining process is done through the use of the AND operator;
2. Process the new query (as it was described in the previous section);
3. If the result is null, the queries are not related and they shouldn't be joined. If the result is non-null, there is a possibility that the new query is a refinement of the old one and the system should give the answer for the query with and without the joining process.

As an example suppose the user makes the following query:

- Accident

After the system's answer, the user asks:

- Drugs

At this point the system tries to join the queries obtaining:

- Accident AND drugs

As the result is non-null, the system answers to both possible queries (drugs; accident AND drugs), allowing the user to select the desired documents.

Using this approach, we are able to infer and to anticipate possible user queries and to act in a more friendly way.

4. Modeling Legal Knowledge

In our project we also intend to be able to model legal knowledge. In order to handle this problem we are representing the legislation using logic programming and we are using Prolog as an inference engine.

As an inference engine we first tried to use YSH, an engine from the AustLLI project (Greenleaf, Mowbray, and van Dijk, 1995) but we needed a more powerful one which, for instance, could be able to model non-monotonic reasoning.

At this phase we have only a prototype representing the legislation that defines when a person has a right for a pension for exceptional services.

As future work we intend to be able to test if a specific document satisfies a specific legal law. In the logic programming framework this task will be done trying to prove the legal rule top-goal. In order to prove the top-goal we will need to transform the rule queries into a semantic representation (for instance DRS) and to check if the document entails the question. At this moment we don't have a semantic representation of the documents, so we can only use boolean search to answer the rule queries.

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. This example is simplified and the bottom goals should be tested against the desired documents.

```
pension(X) <- article31(X), art32(X).

art31(X) <- action_exceptional(X), art31a(X).
art31(X) <- action_exceptional(X), art31b(X).

art31a(X) <- action_exceptional_war_place(X).
art31a(X) <- action_abnegated_and_courageous(X).
art31a(X) <- high_service_country_or_humanity(X).

art31b(X) <- injured_or_deceased(X), act_of_humanity(X).
art31b(X) <- injured_or_deceased(X), act_of_dedication_to_public_cause(X).

art32(X) <- has_respect_individual_colective_rights(X), respect_dignity_of_country(X).

action_exceptional(X) <- action_benefits_country(X,A),
                         action_correct_typology(X,A),
                         action_without_remuneration(X,A),
                         action_beyond_duty_of_functions(X,A).

action_correct_typology(X,A) <- action_serves_national_interests(X,A),
                                action_pressuposes_high_availability(X,A).

action_exceptional_war_place(X) <- action_in_war_place(X,A),
                                   action_beyond_standard_military_patterns(X,A).

action_beyond_standard_military_patterns(X,A) <-
                         action_exceptional_by_military_administration(X,A).
action_beyond_standard_military_patterns(X,A) <-
                         action_defends_other_lives_above_his_own(X,A).
```

If we want to test these inference rules over a specific document, we should try to prove the top goal: pension(X). In order to prove this goal it will be necessary to prove if the person has made an exceptional action which, for instance, needs to be an action which benefits the country. As we don't have a semantic representation of the facts we just try to search the document for expressions that can describe actions which benefit the country. Some examples are:

```
action_defends_other_lives_above_his_own(X,A)<- query "saved live"
action_beyond_standard_military_patterns(X,A)<- query "beyond his duty" and "military".
action_exceptional_by_military_administration(X,A)<-
                         query "beyond his duty" and "military" or "administrat*"
action_in_war_place(X,A)<- query "war"
```

As it can be seen this is a very difficult task, which needs a very good domain description. This task will be done for some legal knowledge during the duration of the PGR project.

5. Bibliography

- [AKPT91] James Allen, Henry Kautz, Richard Pelavin, and Josh Tenenberg. Reasoning about Plans. Morgan Kaufman Publishers, Inc., 1991.
- [AP96] José J. Alferes and Luís Moniz Pereira. Reasoning with Logic Programming, volume 1111 of Lecture Notes in Artificial Intelligence.Springer, 1996.
- [Bra90] Michael Bratman. What is Intention?, in Intentions in Communication. MIT, 1990.
- [CL90a] P. Cohen and H. Levesque. Intention is choice with commitment. Artificial Intelligence, 42(3), 1990.
- [CL90b] Philip Cohen and Hector Levesque. Persistence, Intention, and Commitment, in Intentions in Communication, pages 33--70. MIT, 1990.
- [Car88] Sandra Carberry. Modelling the user's plans and goals. Computational Linguistics, 14(3):23--37, 1988.
- [GS86] Barbara Grosz and Candice Sidner. Attention, intention, and the structure of discourse. Computational Linguistics, 12(3):175--204, 1986.
- [HM87] S. Hanks and D. McDermott. Nonmonotonic logic and temporal projection. Artificial Intelligence, 33, 1987.
- [KR93] 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.
- [KS86] Robert Kowalski and Marek Sergot. A logic-based calculus of events. New Generation Computing, 4:67--95, 1986.
- [LA87] D. Litman and J. Allen. A plan recognition model for subdialogues in conversations. Cognitive Science, (11):163--200, 1987.
- [LA91] Alex Lascarides and Nicholas Asher. Discourse relations and defeasible knowledge. In Proceedings of the 29th Annual Meeting of ACL, pages 55--62, 1991.
- [PR93] J.Pinto and R.Reiter. Temporal reasoning in logic programming: A case for the situation calculus. In D.S. Warren, editor, Proceedings of the 10th ICLP. MIT Press, 1993.
- [Per90] Raymond Perrault. An Application of Default Logic to Speech Act Theory, in Intentions in Communication, chapter 9, pages 161--186. MIT, 1990.
- [Per91] F.Pereira and M.Pollack. Incremental interpretation. Artificial Intelligence, 50:40--82, 1991.
- [Pol90] Martha Pollack. Plans as Complex Mental Attitudes, in Intentions in Communication, chapter 5, pages 77--104. MIT, 1990.
- [QL92] P. Quaresma and J. G. Lopes. A two-headed architecture for intelligent multimedia man-machine interaction. In B. de Boulay and V. Sgurev (eds). Artificial Intelligence V - methodology, systems, applications. North Holland, 1992.
- [QL95] 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.
- [RL92] IrenePimenta Rodrigues and José GabrielPereira Lopes. Discourse temporal structure. In Proceedings of the COLING'92, 1992.
- [RL93] IrenePimenta Rodrigues and José Gabriel Lopes. Building the text temporal structure. In Progress in Artificial Intelligence: 6th Portuguese Conference on AI. Springer-Verlag, 1993.
- [RL97] IrenePimenta Rodrigues and José Gabriel Lopes. AI5, An Interval Algebra for the temporal relations conveyed by a text. In Mathematical Linguistics II, Eds Carlos Martin-Vide, John Benjamins, 1997.
- [Son91] F.Song. A Processing Model for Temporal Analysis and its Aplication to Plan Recognition. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 1991.