

# A Large Case-Based Reasoner for Legal Cases<sup>1</sup>

Rosina Weber-Lee  
Ricardo Miranda Barcia  
Marcio C. da Costa  
Ilson W. Rodrigues Filho  
Hugo C. Hoeschl  
Tania C. D'Agostini Bueno  
Alejandro Martins  
Roberto C. Pacheco

Federal University of Santa Catarina  
LIA - Production Engineering  
rolee@eps.ufsc.br

**Abstract.** In this paper we propose a large case-based reasoner for the legal domain. Analyzing legal texts for indexing purposes makes the implementation of large case bases a complex task. We present a methodology to automatically convert legal texts into legal cases guided by domain expert knowledge in a rule-based system with Natural Language Processing (NLP) techniques. This methodology can be generalized to be applied in different domains making Case-Based Reasoning (CBR) paradigm a powerful technology to solve real world problems with large knowledge sources.

## 1. Introduction

Court decisions stored in large databases need efficient retrieval mechanisms to improve jurisprudence research. We have been working with a database that contains 90,000 legal texts (court decisions). The bottleneck is to convert these texts into cases to build the CBR system's case base from where the cases can be retrieved. We present a knowledge-based methodology to convert legal texts into cases, showing that it is possible to automatically model cases from texts. We demonstrate this within the specific domain of decisions of a State Court in Brazil. The fact that these legal texts are highly stereotypical is one of the reasons of the accomplishment of such task. Although the key principle is knowledge -- domain expertise, this is what makes feasible the automatic information extraction of these texts.

Our project refers to a large case-based reasoner in which the task is to provide the user with the most useful cases to support one legal input problem. The intended contribution is to show how to create case-based reasoners to solve real world problems that require large case bases.

---

<sup>1</sup> In Proceedings of the Second International Conference on Case Based Reasoning, 190-199. David Leake, Enric Plaza (eds.), Providence, RI, Berlin: Springer.

### **1.1 Background**

The attempts in developing intelligent systems in the domain of law were boosted by HYPO -- a CBR system that creates legal arguments from a case base on the domain of trade secret law (Ashley & Rissland, 1988a,1988b). In this program, dimensions are used to dynamically perform indexing and relevancy assessment of past cases. Most importantly, the system demonstrated how to handle arguments and lessons present in legal cases. However, the hand-coding required in the development of such systems prevented them from becoming a paradigm for real world problems.

Considering different approaches to the automatic treatment of legal texts and documents, Branting and Lester (1996) describe the design task of document drafting that requires complex adaptation for case reuse. In their approach, they demonstrate the illocutionary and rhetorical structures of self-explaining documents.

Daniels and Rissland (1995) built a hybrid CBR and Information Retrieval (IR) system where the CBR technology plays the role of improving the query presented to the IR system, improving the results. This alternative stems from their claim that texts are not amenable to knowledge-based methods, therefore they do not benefit from the ability of retrieving relevant cases what is the main strength of CBR technology.

SALOMON (Uyttendaele, 1996) project proposes the improvement of access to legal texts through automatically generating summaries of court decisions of criminal cases. The summary represents court decisions through nine attributes in which the values are extracted from the texts. The approach combines statistical and knowledge-based techniques. Although it deals with indexing, they do not explore CBR technology.

All these efforts point to the necessity of developing tools to link the CBR usefulness in retrieval and real world domains.

### **1.2 Goals**

The main goal is to develop a large retrieval-only case-based reasoner to retrieve the most useful cases to support jurisprudence research. In order to accomplish this goal we propose a knowledge-based methodology to automatically convert legal texts into cases. The intended result is to obtain cases modeled with descriptors extracted from legal texts. One major benefit from the development of large CBR systems in the legal domain is to make possible reusing the knowledge embedded in jurisprudence that is used to reference new court decisions. Improving the access of past legal cases enlarges the horizon from where new decisions are grounded, consequently raising the quality of the results of the judicial system.

Next section presents the domain of the application. Then we introduce our CBR system project. In section 4, we demonstrate the methodology to convert texts into cases. Finally, the conclusion is in section 5.

## **2. The Domain**

Legal cases along with general principles, laws, and bibliographical references are the traditional sources of Law. Law is predominantly implemented by lawsuits, which refer to disputes of parts that advocate conflicting arguments. The universe of information comprehended by the subject where the conflict takes place is unlimited.

The extent of the information embedded in the subject of lawsuits sometimes goes beyond the human capability of reasoning and understanding. This makes the law domain a fertile area for intelligent-based applications. Artificial Intelligence literature points to several efforts on modeling legal reasoning based on CBR technology as the most appropriate tool to reason within this domain, (Bench-Capon, 1995).

This paper focuses on the application of the CBR technology to retrieve legal cases that describe court decisions. The importance in choosing this part of the domain lies on the fact that these legal cases are referenced as foundations of petitions and decisions.

Brazilian professionals have two sources to search for past legal cases to ground new decisions: books and database systems. The available database systems consist of data from abstracts of legal decisions. These systems are limited to a recall that provides around 25% of relevant cases, (Blair, 1985).

Facing this dearth of resources, we are pursuing a system that makes feasible the search for relevant legal cases enlarging the reach of the research results. Achieving such goal will enhance the tasks of these professionals, representing an improvement to the judicial system in contributing to a better society.

The State Court of Justice (SCJ) records have around 90,000 machine readable complete descriptions of legal cases (not only the abstracts). These descriptions are the basic entity of our application. They describe the experiences that are the cases in the CBR system. Our methodology to convert these legal texts into legal cases is presented in section 4.1.

### **3. The CBR Application**

The project's final purpose is a retrieval-only CBR system that plays the role of an intelligent researcher. The task to be performed by the system is the same as judges and their assistants perform: search for legal cases in the jurisprudence. The human experts guide the search by some hint they might have about the case but there are no effective means to guarantee the efficiency and the reach of the results. This CBR system aims at guaranteeing an efficient search, that is providing useful recalls. Also, the system aims at providing a better reach in terms of ensuring that every relevant case is actually recalled.

Initially we have developed a prototype using only court decisions on *habeas corpus* petitions in homicide crimes to evaluate the potential of a case-based reasoner to retrieve legal cases. The descriptors that indexed the cases were chosen attempting to capture strengths and weaknesses of the texts to provide usefulness to the retrieval. The hand-coded initial descriptors are the following: manslaughter (Boolean), qualification (list), status (list), conspiracy (Boolean), application (list), legal foundations (list), subject foundations (list), arguments (list), unanimity (Boolean) and result (list), as well as identification descriptors such as data, place and reporter; petition type and category were default to *habeas corpus* and homicide.

The response from legal experts motivated us to develop a reasoner able to embody all types of legal decisions. The legal experts suggested relevant descriptors and some features to the interface. They have also suggested a feature to perform new retrievals based on a smaller set of descriptors to be chosen by the user. The

requirements of domain expert knowledge became evident in the development of many CBR problem areas. The implementation of the reasoner is essentially guided by expert domain knowledge.

petition type: <i>habeas corpus</i>		
reporter: <i>Des. José da Silva</i>		
city: <i>Lages</i>		
number: <i>10.282</i>	page: <i>06</i>	date: <i>25/03/92</i>
category: <i>homicide</i>		
result: <i>accepted</i>		
unanimity: <i>yes</i>		
active part: <i>defense attorney</i>		
passive part: <i>district attorney</i>		
subject foundation: <i>insanity</i>		
foundation concerning procedures: <i>annulling</i>		
application: <i>abatement</i>		
laws cited: <i>articles 26 &amp; 97 of Penal Code</i>		
argument 1: <i>first offender</i>	argument 2: <i>non compos mentis</i>	
argument 3: <i>negligence</i>	argument 4: <i>circumstantial evidence</i>	
manslaughter: <i>yes</i>		
qualification: <i>simple</i>		
status: <i>consummated</i>		
conspiracy: <i>no</i>		
corpus delicti: <i>yes</i>		

Figure 1. Surface features and dimensions representing a case.

The prototype described above gave us an idea of the type of descriptors required to index cases to retrieve relevant legal cases. In the new reasoner, we use two types of descriptors: surface features and dimensions. Surface features are easily assigned from a specific part in the text. Although initially inspired by HYPO's dimensions we use this term to refer to the attribute<sup>2</sup> part of a descriptor, whose values have to be inferred from the text with the use of domain knowledge. A case in the new reasoner is modeled with the initial descriptors according to Figure 1.

The dimension *category* represents a delict (if within the criminal area) or the subject of the lawsuit. Law categorization is provided by the national penal and civil codes (and other sources of laws) that we have represented through a *tree of categories*. When filling out the input case, the end-user provides a category that may be at any level of the tree. The intelligent interface identifies the category in the tree and shows the upper levels to the user asking for confirmation, (see Figure 2). Suppose the end-user enters *theft*, the interface shows: level 1: *criminal*; level 2: *crimes against property*; level 3: *theft*, that represents the categorization according to the Brazilian criminal code.

---

<sup>2</sup> Dimensions are called the attribute part of an attribute-value pair of a descriptor, (Kolodner, 1993, chapter 9).

The main result of the development of this large CBR system equals furnishing a human expert with the memory capacity and speed of a computer.

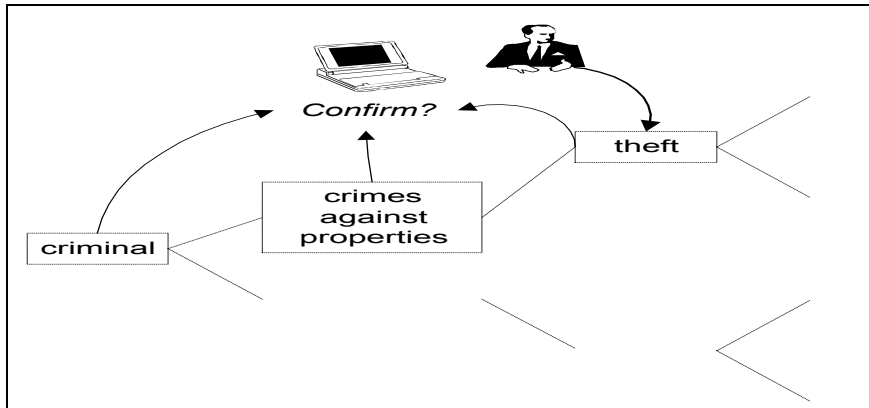


Figure 2. The interface searches for the categorization in a tree of categories

#### 4. Legal Cases

The knowledge representation in a large CBR system is the main issue to be resolved. The cases that comprise the case memory describe legal texts. These texts are converted into cases through the definition of attribute-value pairs that describe and index the cases. Next, we describe a methodology to perform this conversion automatically.

##### 4.1 Methodology

The methodology to convert texts into cases was developed with the knowledge elicited from domain experts. There are two distinct phases, the development of the methodology and its implementation. Figure 3 illustrates these two phases and the steps they comprehend. Next, these steps are described regarding the development and implementation phases.

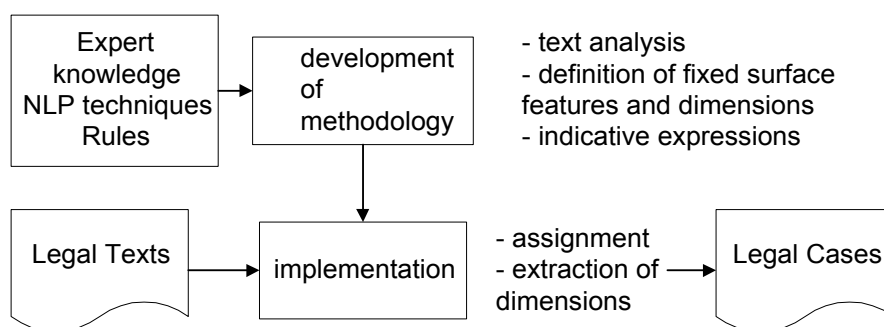


Figure 3. Phases and steps of the methodology.

##### 4.1.1 Text Analysis

The development of the text analysis is required only at the beginning of the process when the domain is chosen. The type of texts from where the cases will be converted are analyzed by domain experts. The goal is to define the rhetorical structure of the

texts and identify the parts in which the illocutionary expressions are present. We have performed sample tests to ensure that each substructure is actually present in every legal text. The rhetorical structure of the legal texts is described through the following substructures.

1. Identification: surface features such as date, city, reporter and petition type.
2. Abstract: varies in its length, starts after the end of the *identification* and ends with two paragraphs, the first indicates the applicant and the second presents the result.
3. Body: in its conclusion it is usually the court decision and its foundations. This is where the search for illocutionary expressions takes place. Upper paragraphs describe details of the situation, indicating the laws that categorize the subject, and points to foundations.
4. Closing: starts with one paragraph about votes followed by date, place and names of participating attorneys.

The implementation of text analysis consists of running a Natural Language Processing program that reads legal texts and identifies parts of the text that will be used in other steps.

#### 4.1.2 Definition and Assignment of Fixed Surface Features and Dimensions

The development of this second step started with the knowledge elicitation from domain experts who have defined a small set of attributes to describe all legal texts. Experts expect them to be valued in all cases. These attributes are illustrated and exemplified in Figure 1.

<i>surface features/dimensions</i>	<i>substructure</i>
petition type	identification
reporter	identification
date	identification
city	identification
number	identification
page	identification
category	abstract and body: categorization
outcome	abstract: result
active part	abstract: applicant paragraph
passive part	abstract: applicant paragraph
application	abstract and body
foundation concerning procedures	body
subject foundation	body: conclusion
legal foundation	body: conclusion
arguments	body
first offender	body
laws cited	body: categorization

Table 1. Position of attribute values in the structure of the legal text.

It is important to point out that the definition of the attributes do not require the experts to examine a significant amount of texts. Their capability of pointing out

these attributes relies on their expert knowledge of the domain. Next, experts were asked to point the substructure where the value of each attribute is informed. Results are shown in Table 1.

The knowledge acquisition process elicits from experts how the values appear in the texts. Rules were developed to be applied on each substructure to extract values for the attributes. The resulting rules are programmed in Prolog in a Natural Language Processing system that reads the substructure and assigns values to the attributes.

Given an attribute, the rule-based system is supposed to find its proper value. These are some guidelines to accomplish the task:

- identify in what part of the text to apply the rule, (it may be more than one part);
- check the existence of any useful information relevant in the dimensions and features already assigned;
- check the existence of general domain knowledge from where to obtain useful information, e.g., the tree of categories;
- check the existence of a limited list of possible values to assign to a dimension;
- identify the format of the value (e.g., list, number, sentence; single or multiple, etc.) if one is recognized.

These guidelines orient rules and strategies that are employed by the system. One of the strategies employed in this search is the use of word lists. The sentences within the proper paragraph are represented by lists of words and the system checks whether certain words or combinations of words occur in the lists. Let us illustrate this process with the assignment of the dimension *outcome*, that represents the result for the petition. The first requirement for the rules related to the result is the petition type; because depending upon it, the result may be expressed with different terms. For instance, in petitions for *habeas corpus*, the verb used to express its acceptance is '*conceder*' (concede, accept), whereas the verb '*denegar*' (refute, reject) is used to reject the petition. In different types of petitions, other verbs are employed to express acceptance, such as the verb '*prover*', which is a synonym of accept although it is not used in certain types of petitions. This information is obtained by the knowledge acquisition step. It narrows the problem in a such a way that we can draw rules as, "If petition type is *habeas corpus* then search in the substructure abstract:result for the verbs '*conceder*' and '*denegar*'". In Figure 4 the interface<sup>3</sup> shows two instances of substructure abstract:result where the outcome is informed. In Figure 5, the command '*resultado*' stands for triggering the rules that return the value for the outcome that is '*denegado*' when rejected (10881) and '*concedido*' when accepted (10886). This example demonstrates the use of expert knowledge in orienting the search for the proper values in the text. In assigning values for dimensions involving facts of the domain, the complexity of the rules increase. The system is designed to return a warning if a value is not found. Whenever a new expression is used by a reporter avoiding the system to trigger any rule, the system informs this failure and a new rule is created. This device guarantees efficiency and aids the maintenance of the system. The development phase ends when rules are provided to value all attributes.

---

<sup>3</sup> Amzi!Prolog 3.3Mar96 Copyright ©94-95 Amzilinc.

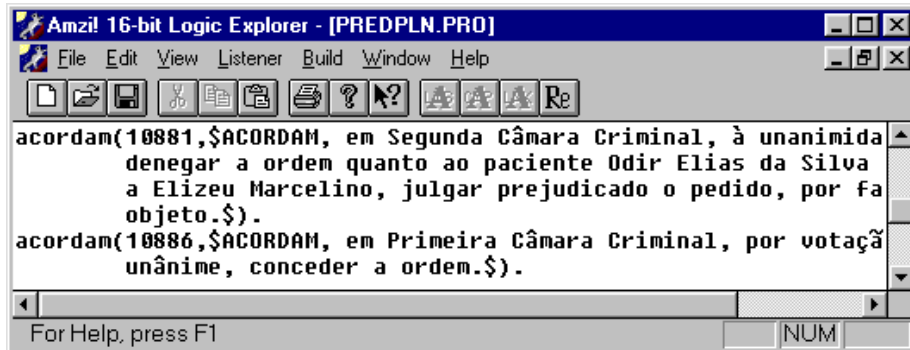


Figure 4. Portions of two legal texts where the outcome is read.

#### 4.1.2.1 Testing Rules

The procedure for testing rules is the same for all attributes. To test the rule set oriented to extract the result of habeas corpus cases, we have gathered 684 texts – referring to all cases of this type from 1990 to 1996. The first rule set stemmed from a sample of 17 texts. Applying this rule set on the 684 texts, generated a 63% rate (434) of proper assignments. Two new rules were added and the execution of this new rule set resulted in 678 assignments. Out of the 6 cases left without values, 5 of those referred to cases where no result has been decided – the court relegated the decision to another court; only one (1) case provided no information about the decision in the substructure abstract:result. We consider this 99% (678 out of 684) good enough.

The implementation of the assignment phase can be performed since all rules are tested. In this phase, the rule-based system receives the texts and assigns values for all surface features and dimensions.

At this point, we already have cases to the reasoner, however we understand that case descriptions can be improved, and this is what we pursue with the dynamically extracted dimensions.

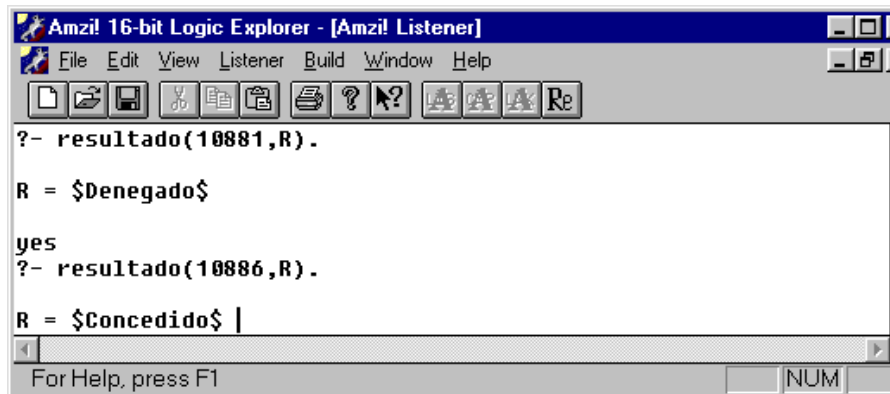


Figure 5. Results of the reading of the value for the outcome in two legal texts in Amzi!Prolog.

### 4.1.3 Extraction of Dimensions

The analysis of samples of legal texts by domain experts resulted in the observation of some repeated expressions that were usually present indicating relevant information about the case. This is how the development of this step has come up, with the identification of *indicative expressions*. These expressions were somehow connected to lessons provided by the cases. In a first evaluation, we have selected the following four types of indicative expressions.

1. Nouns derived from adjectives: they express a condition; e.g., impossibility.
2. Sentences with verb to be in which the noun is an object of the domain: express an information about the state of the object; e.g., custody is, victim has been, evidence was, perpetrator is, defendants were, etc.
3. Verbs that are objects of the domain: indicate facts; e.g., certify, arrest, allege, prove, etc.
4. Adverbs meaning *for this reason*, indicate conclusive lessons; e.g., therefore, ergo, hence, thus, therefore, accordingly, consequently, so.

When experts were asked about how to use these expressions, they suggested heuristics. One example is the noun *impossibility*. According to experts, nouns derived from adjectives indicate the presence of a lesson. Suppose the legal case reads, "...the impossibility of penalizing the defendant stems from the fact the defendant is under legal age and therefore his imprisonment constitutes a misfeasance...". This sentence clearly teaches the lesson that the defendant who is under age cannot be kept imprisoned. The sentence following the expression impossibility will usually inform about an illegal fact, whereas the sentence following therefore can either inform an illocutionary expression or expose reasons for the assertions, i.e., reveal the grounds for such impossibility. From this fact we can determine another dimension concerned to the grounds of the condition. Hence, the dimensions extracted from this first example would be: *penalizing condition* and *penalizing condition grounds*; and the values would be respectively *impossible* and *defendant is under legal age*. These observations usually hold, guiding the definition of heuristics and rules to extract dimensions. Another instance is in the text, "...It is characterized the inadmissibility of the evidence when the means for its acquisition are illicit." The resulting dimensions are evidence condition and evidence condition grounds.

The implementation of this phase is accomplished first with a search for indicative expressions throughout the text, followed by the employment of heuristics to extract and value the new dimensions. Suppose the system finds the sentence, "The subpoena was annulled because the defendant was not properly served." It is a sentence with the verb to be and the noun is an object of the domain. The heuristics indicate that this sentence informs about the state of the object. The new dimension to be created is *state of subpoena* and its value is *annulled*.

These dynamically extracted dimensions are defined exclusively in the cases in which the originating text contain them. Hence, during a search in the reasoner, if many of the retrieved cases present some type of dynamic dimension, the value for the dimension is asked to the user in order to improve retrieval. If the user has no information or there are not many instances of a given dimension, then it is not

included in the similarity assessment because the purpose of these dynamic dimensions is only to improve similarity.

## 5. Conclusions

It has been demonstrated an approach to automatically convert texts into cases to minimize the bottleneck of knowledge representation in large CBR systems. This approach can be generalized to be applied in different domains making CBR a powerful technology to the treatment of real world problems with large knowledge sources.

The rule-based text classification techniques employed in the very restricted domain of knowledge have limitations and advantages. The expert knowledge orientation makes it necessary a knowledge elicitation step. On the other hand, the limited domain guarantees that once the rules are elicited, they work for most texts, what has been demonstrated by our first experiments.

The development of a large retrieval-only CBR system in the domain of Law represents a solution to the search for jurisprudence, improving the quality of its results.

## 6. References

- Ashley, Kevin D. & Rissland, Edwina L. (1988a). Compare and Contrast, A Test of Expertise. *Proceedings of a Workshop on Case-Based Reasoning*, 31-36.
- Ashley, Kevin D. & Rissland, Edwina L. (1988b). Waiting on weighting: A symbolic least commitment approach. *Proceedings of AAAI-88*. Cambridge, MA: AAAI Press/MIT Press.
- Bench-Capon, T.J.M. (1995) *Argument in Artificial Intelligence and Law*. JURIX 1995.
- Blair, D.C. & Maron, M.E. An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. *Communications of the ACM*, 28 (3), 289-299, March 1985 in Daniels & Rissland, 1995.
- Branting, L. Karl & Lester, James C. (1996) Justification Structures for Document Reuse *Advances in Case-Based Reasoning: third European Workshop; proceedings/ EWCBR-96*, Lausanne, Switzerland, November 14-16, 1996. Ian Smith; Boi Faltings (ed.)-Berlin; Springer,1996.
- Daniels, J. J. and Rissland, E. L. (1995). A Case-Based Approach to Intelligent Information Retrieval. *Proceedings of the SIGIR '95 Conference SIGIR '95 Seattle WA USA 1995 ACM*.
- Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann, Los Altos, CA.
- Klahr, Philip (1996). Global Case-Base Development and Deployment. *Advances in Case-Based Reasoning: third European Workshop; proceedings/ EWCBR-96*, Lausanne, Switzerland, November 14-16, 1996. Ian Smith; Boi Faltings (ed.)-Berlin; Springer,1996.
- Uyttendaele, Caroline, Moens, Marie-Francine & Dumortier, Jos. SALOMON: Automatic Abstracting of Legal Cases for Effective Access to Court Decisions. *JURIX 1996*.