# On Legal Texts and Cases



Rosina Weber
rosina@eps.ufsc.br
Rua Maestro Aldo Krieger, 138/403
Florianópolis, SC 88.037-500 BRAZIL
phone: 55-48-234-6574
Alejandro Martins
martins@eps.ufsc.br
phone: 55-48-331-7055
Ricardo M. Barcia
rbarcia@eps.ufsc.br
phone: 55-48-331-7008
Federal University of Santa Catarina, BRAZIL
Graduate Program of Production Engineering
Laboratory of Artificial Intelligence
Campus Universitário -Trindade
Caixa Postal: 476 CEP 88.010-970
Florianópolis - SC

**Abstract**

The search employed by judicial professionals when seeking for past similar legal decisions is known as jurisprudence research. Humans employ analogical reasoning when comparing a given actual situation with past decisions, noting the affinities between them. In the process of being reminded of a similar situation when faced to a new one, Case-Based Reasoning (CBR) systems simulate analogical reasoning. Judicial professionals have two sources of jurisprudence research: books and database systems. The search in books is time-consuming and imprecise due to the limitations of humans' memory. Available text database systems do not guarantee the retrieval of useful documents. PRUDENTIA is the case-based reasoner tailored to the Brazilian system that confers efficiency to jurisprudence research. Judicial cases are described with natural language text, comprising a collection of textual documents. These texts are the experiences that require case engineering to be modeled in a structured representation of cases. We have developed an automatic means of performing the case engineering, that is, converting legal texts into structured representation of cases. Examples of PRUDENTIA demonstrate the power of similarity-based retrieval in a textual CBR system against text database applications improving the usefulness of the documents retrieved.



## 1. Introduction

The issue of textual case-based reasoning comes up when textual documents contain descriptions of experiences of a given domain knowledge. A case should express an experience, but cases also have to be manipulated within the Case-Based Reasoning (CBR) architecture. So, textual documents have to be represented as cases; texts should be referred to as the original source of description of an experience, whereas cases are the entities in a case-based reasoner that permit experiences to be manipulated and retrieved. Allowing the proper handling of such experiences demands a more structured representation of the experiences.

Each experience, or class of experiences, has to be evaluated in order to be properly associated with the goal task in a CBR system. Medical experiences might be related to diagnostic tasks, while legal experiences may relate to interpretation or classification. Once the reasoner's task is defined, a knowledge engineer can evaluate whether one or more classes of experiences should be used as the knowledge source of the system. After defining the scope and the task of the system, it is easier to envision a proper structure to represent the chosen experiences.

Knowledge engineering deals with the acquisition and representation of domain theory, rules of thumb and any available knowledge that human experts can provide and express. The way of facing an experience within a given domain knowledge in terms of what to highlight in a case representation is a controversial and polemic issue. Narrowing such abstraction into more specific problems is an appropriate guideline. Hence, let us discuss the viewpoint of representing experiences with

cases within the legal domain and figure out some lessons that may be extended to other fields.

The distinction between domain theory and facts of life has been pursued by several authors in Artificial Intelligence and Law (AI&Law) such as Smith (1987, 1997a, 1997b), Branting (1991), Valente (1995). Trying to discriminate between these two different aspects in legal experiences has not proved efficient, as most of these researchers are still searching for approaches that enable intelligent legal systems to succeed in real world applications. Hence, we suggest to interpreting the legal experiences in a very human fashion in instead of separating domain theory from other aspects. Judicial professionals are the experts that interpret and deal with legal experiences. The lawyers' expertise is indeed interpreting legal texts that describe legal experiences. Thus, this is the ability that might be represented in an intelligent legal system: the lawyers' expertise in interpreting legal experiences. Consequently, the case structure should follow guidelines that comprise the legal expert's standpoints of the legal experiences, and the knowledge about this interpretation is the object to be elicited.

Once the structured representation of a case is obtained through a knowledge engineering effort, the problem results in mapping the textual experiences that embed the case content into the structured representation of cases. This conversion is what enables the development of the PRUDENTIA system, an interpretive case-based reasoning system that retrieves the most useful cases to support jurisprudence research. The ultimate goal of this project is to provide judicial professionals with an intelligent research tool enabling a quick and efficient judicial system. Cases in this reasoner are descriptions of legal decisions that are originally written in natural language text. PRUDENTIA searches for legal situations that can be useful in teaching lessons to a new situation. The system returns similar situations that are found through analogical reasoning simulated by the CBR inference. This current version of PRUDENTIA comprises descriptions of 3,500 legal decisions that represent knowledge source of jurisprudence research. The case-based reasoner performs analogical reasoning, comparing a new legal situation to the legal decisions in the case base, and returns a set of similar situations.

## 2. Mapping Texts into Cases in the Legal Domain

Our specific problem is mapping a textual description of a legal experience into a structured representation under the guidelines provided by expert knowledge. The guidelines impose goals and constraints to keep the structured representation consistent with the expert interpretation of a legal experience. In practical terms, the correct representation must result in legal experts comprehending the same content when reading the textual description as when reading the structured representation.

Employing expert guidelines does not exclude known CBR guidelines to define an indexing vocabulary. On the contrary, they transcend them. One basic requirement is that the expert is able to envision the whole collection of experiences as to anticipate values for every characteristic. Let us describe the methodology applied in the development of the PRUDENTIA system.

The system performs the same task as judicial professionals when searching for legal cases in jurisprudence. When this task is performed by human experts, they conduct the search by comparing an interpretation of a given legal situation to interpretations of the legal descriptions. Experts seek for similar legal situations that might provide insights to the new situation. Human experts employ reasoning by analogy (Durkin, 1994) when performing this task. Therefore, CBR is the appropriate technology to accomplish the task of research, as the only intelligent paradigm that simulates analogy. The result of the development of this large CBR system is equivalent to furnishing a human expert with the memory capacity and speed of a computer.

Jurisprudence research is employed and required in several activities within the legal domain. The legal profession embodies different activities ranging from adjudication and consultation to legal administration and education. Predominant activities of judicial professionals can be categorized into different fields of legal activity: legal planning, argumentation, adjudication, legal management, legal analysis, teaching and legislation, among others.

In the legal activity of adjudication, judges are subject to a methodology supported by laws to building sentencing. In this methodology, one of the requirements is that judges must use jurisprudence research as part of the process of stating and supporting their decisions. Meanwhile, defense attorneys and prosecutors attempt to prove their points, laying the groundwork of argumentation on jurisprudence. Within most of these activities, jurisprudence research stands as a relevant tool that augments the correctness of every task. Effective jurisprudence research promotes a just society.

The focus of our research is Brazilian jurisprudence. The Brazilian system of law is civil law, which is derived from Roman law and which is practiced in most European countries. Civil law has organized codes as the main source of law. This systems differs from the common law system of America and England in which the basic source is case-by-case judicial decisions. In the Brazilian system, decisions are consequently one source of the Law but it is not the most important one.

As the source for legal decisions, we are focusing on legal decisions produced for criminal appeals by the State Court of Justice (SCJ) of Santa Catarina - an intermediate appellate court - in the period from 1990 to 1996. SCJ records from this period consist of 17.2 Mb of 3,447 machine readable complete descriptions of legal

cases (not only abstracts). These records comprise $2.5x10^6$ words, with $10^7$ characters. These descriptions are the basic entity of our application. They describe the experiences that are the cases in the CBR system.

## 3. Mapping Legal Decisions into Cases in PRUDENTIA

The presence of stereotypical substructures in the legal texts facilitates the process of performing the automatic mapping of the experiences into cases. Branting & Lester (1996) suggested rhetoric structures of legal documents what oriented us in the definition of substructures that can easily relate to some aspects of each experience. Experts can associate each substructure to some important information that can be used to value features in a formlike representation of cases.

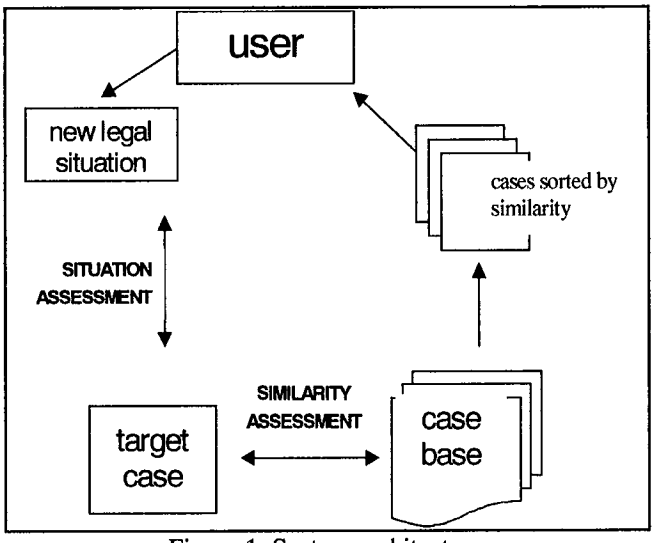


Figure 1. System architecture.

The conversion of textual experiences into structured representations of cases is performed through the steps of organizing the textual experiences in functional substructures and associating these substructures with features in a formlike case representation. Since we have small parts of texts where some information can be extracted, the problem is no longer CBR related, but from this point is instead a natural language problem.

PRUDENTIA's basic architecture is laid out in Figure 1. The inference starts with the identification of a new legal situation. This occurs when a judicial professional performing usual legal activities encounters a new legal situation that requires jurisprudence research. The legal professional starts a session in PRUDENTIA with an interpretation of this new legal situation in mind. The system attempts to elicit the new legal situation from the user's mind through the process of situation assessment. Situation assessment methods infer values to assign the attributes in the formlike representation of system's cases, modeling the new legal situation in the same fashion as cases in the case base. The system then compares the new legal situation - henceforth referred to as target case - to every candidate case in the case base. A similarity metric measures the value of each similarity that is used to sort candidate cases to comprise the outcome of an iteration. The next sections present the participating processes and issues in the implementation of PRUDENTIA.

The case base comprises the collection of cases and the mechanisms used to connect cases to the architecture. Organizational structure in CBR theory refers to the way cases are organized in the case base. In PRUDENTIA we make use of a formlike representation of cases that are organized in a flat structure. The flat organization of cases is implemented through a relational database, allowing a great amount of cases in the case base. Cases in this reasoner are modeled with a formlike representation, i.e., a set of fields (attributes) properly valued.

The case engineering problem starts from a textual description of a legal experience that has to be mapped into a structured representation under the guidelines provided by expert knowledge. The guidelines impose goals and constraints that keep the structured representation in accordance with an expert interpretation of a legal experience. In practical terms, the correct representation must result in most legal experts comprehending the same content when reading the textual description as when reading the structured representation.

| | |
|---|---|
| petition type | The entire scope of the current domain comprises around 200 petition types that vary in occurrence and importance. *Criminal appeals, cross-appeals,* and *habeas corpus* are examples. |
| number | The number given to order legal decisions. |
| reporter | The name of the reporter who issues legal decisions. |
| district | The district where the original act that triggered the lawsuit has taken place. |
| page | Localizes the decision in the textual file. |
| date | Date of decision. |
| foundation(1,n) | Foundation is the basis on which an appeal is founded. |
| theme(1,n) | Secondary legal issues and circumstances. |
| secondary laws | Secondary laws that may be brought up to support formal actions. |
| category | The charges (felony or misdemeanor) that correspond to an article of a law or to the Constitution. |
| result | The court decision, either positive, negative or neutral. |
| unanimity | Unanimous decisions are not eligible to be reviewed by a higher court. |

Table 1. Attributes in PRUDENTIA.

The formlike representation consists of a set of attributes that embody the content and context of the

experience in which the knowledge will be conveyed. Every attribute properly valued represent a descriptor that supports case representation. Attributes in PRUDENTIA are: petition type, number, reporter, district, page, date, foundation *(1,n)* [for *1* to *n* values], theme *(1,n)*, secondary laws, category, result, unanimity. These attributes are described next in Table 1.

The indexing vocabulary is the essence of retrieval. *Essential* indexes are those that are indispensable in guaranteeing similarity assessment success and consequently ensuring an efficient retrieval. *Foundation* and *theme* comprise the set called essential indexes. The qualification proposed for such indexes stems from their nature and relevancies in retrieval.

Basic indexes consist of values for attributes *category* and *petition type* and at least two values of essential indexes. The basic indexes constitute the minimum subset of indexes needed to start the situation assessment process. Hence, we consider these basic indexes as the minimum values to represent an experience within the current context.

## 4. Natural Language, Information Extraction and Template Mining

Text-based applications of Natural Language Understanding deal with several issues, among them, the one of extracting information from texts. Information Extraction (IE) is the area concerned with extracting specific types of information from large volumes of unrestricted text containing information in some domain (Lehnert, 1993, 1994, 1996). An IE system must be input with domain guidelines that specify what to find and what to extract. Within this field of knowledge, one very simple and easy-to-use technique that has proven efficient is Template Mining.

Template Mining is a NLP technique that extracts data from texts when the text forms recognizable patterns from the target to be extracted or its surroundings. A template carries information on what to search in the text and it is triggered to extract the parts indicated (Lawson et. al., 1996).

Employing template mining to the rhetorical substructures defined in the textual documents represents a solution to the conversion of legal textual experiences into structured cases. Template mining methods search the substructures of the texts extracting values that are used to ascribe features in the structured cases, representing and indexing them. Next, we briefly describe the methods employed to extract values to the following attributes.

### 4.1 Extraction of Values for Indexing Assignment

Cases in PRUDENTIA are represented through a formlike representation that comprises the attributes as described above. Most attributes are single, i.e., receive one single value, while indexes theme and foundation can have multiple (1,n) values. The methods that make possible the indexing assignment of each attribute are presented next.

The value for *petition type* is extracted directly without any expert knowledge because the second line in the heading of legal texts begin with this value. Only a confirmation that the complete expression is extracted is required. This is performed by checking the list that comprises 204 petition types. This same procedure is executed for other attributes in the substructure heading, that are reporter, date, district, number and page.

*Category* refers to the law that has originated the lawsuit, e.g., a felony or a misdemeanor. The format of the attribute is one of a list of words that represent the title of the article or the law. The possibility to value the attribute with the number of the article or law is future work. Meanwhile, we use the number of the law as a means of eliciting the textual value for the attribute.

The method for *category* starts on the substructure body:categorization. This substructure is a paragraph that usually brings the specific article, law and source. It is written in a sequence following the district where the felony has been committed and after an expression equivalent to "for infringing articles 26 & 97 of Penal Code" (e.g., penal code).

The method extracts the law and its source and translates the information to the title of the category. In the current application, this process has ascribed values in about 2,600 cases. The remaining texts do not contain the law specified and we have to turn to the following stage.

The second process in this method is held by a search on the substructure abstract:main. This second stage searches for the title of the *category* that is fed by the results in first stage, complementing information reused in former versions. This process resulted in another 800 cases with values for *category*. The 100 left were given back to the experts for evaluation. Again, here we face a difficult problem: recognizing when a given value is not present in the original text. This happens at times in appeals that are rejected without detailed considerations. The incremental process continues until every case is properly valued.

Index *foundation* represents legal aspects or material facts that substantiate an appeal or its decision. A different strategy is deployed to extract values for *foundation*: direct search for lists of expressions in the respective substructures. The most relevant values for foundation are usually given in the abstract:main. Others are found in substructure body:conclusion. Therefore, we mine first in abstract:main and afterwards in body:conclusion.

The very interesting hint concluded from the knowledge acquisition is that some specific expressions simply cannot mean anything else but a foundation, especially if they are mentioned in certain portions of the text. Thus, experts come up with heuristics where we can assume that an expression such as first offender or

negligence necessarily indicates a foundation if it appears in those substructures.

However, some expressions have the semantic meaning guaranteed when extracted from the abstract that does not hold for the substructure body. Hence, this method has an intermediate stage that treats some words depending on the substructure they have been extracted from. Examples would be expressions such as blame and different conjugations of the verb to confess that reveal their relevance in that decision if they are mentioned in the substructure *abstract*. These expressions within body may be simply part of an explanation, not necessarily indicating an important issue within the context of the decision.

The attribute *foundation* is multiple-valued. This is because it is an attribute responsible for representing every aspect grounding the appeal. Moreover, we search for expressions in different substructures resulting in a large number of values. However, it is possible that, in some cases, there are only one or two values for foundation.

After extracting a number of values from a collection of cases, experts have noticed the resemblance between some words. The list of expressions had to be reviewed by experts who indicated expressions that work as synonyms within the legal context. For instance, the words jail, custody, prison and penitentiary; and sometimes the verb to arrest. This list of synonyms improves efficiency augmenting the retrieval of useful cases in a human fashion.

In the current prototype, the process started from the reutilization of the list of expressions of the former ones. The incremental process took place until every case was properly valued.

The index *secondary laws* refers to articles of laws that are mentioned throughout the texts. This may happen when a different categorization is pursued or, for instance, when no substantial matter is to be considered due to an annulment caused by some error of law. The law errors are usually indicated by the respective law.

The law articles may indicate arguments used by one of the parts in validating assumptions. The different sources of law demand a two dimensional valued attribute represented at the level of the number of the article or law and the source, such as Article 12 of Federal Constitution.

Values for secondary laws are not searched in any specific substructure, but in the whole text. The method is implemented by first extracting the articles that refer to the categorization, since they are the value of the attribute category. Then, we select only portions of the text that contain numbers. Template mining is used to find the valid sources after expressions, such as article and its variations, through wild cards.

In the substructure abstract, the last paragraph starts with a sentence where the value for *unanimity* can be extracted: abstract:result. In the occurrence of a dissenting opinion, the text reads under majority of votes. There might be at most ten inflections to express such characteristic, making it very simple. The value for unanimity is Boolean, because a decision is either unanimous or not.

There are different fashions to express if a petition has been affirmed or not, and these forms vary in correspondence with the type of petition. The substructure in the texts is very stereotypical. The knowledge acquisition was carried out for the second prototype for habeas corpus and criminal appeals. As the forms of indicating the *result* vary in accord with the petition type, we were able to implement only one method. The method selects the substructure abstract:result and runs a kind of a demo rule verifying the petition type and orienting to a specific knowledge base where the respective list of expressions is.

This is the only method that was implemented in Prolog, as the use of logic programming turned out to be more efficient than the use of wild cards for this attribute. Except for the rules and the design of different knowledge bases, this method is not amenable to template mining for single expressions. The expressions resulting from the knowledge acquisition process yielded expressions that demanded a NLP treatment.

The first requirement for the rules related to the result is the petition type, because the result depends upon the petition type, and 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, affirm, 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*, that 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*".

This example demonstrates the use of expert knowledge in orienting the search for the proper values in the text. 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 index *theme* refers to some secondary aspects or circumstances that characterize cases. The complexity of these indexes stems from the fact that they were defined to complete the universe of the attributes in describing the content and context of the experiences on legal decisions. Values for attribute *theme* that may be present in a legal case can be grouped into classes of the same nature: the class of tests required (mental health evaluation required, evaluation of drug dependency required); the class of application (application for abatement, application for suspension, application for abatement); the class of external context (traffic accident,

strikes, penalty reduction). The assignment of values for this index completes the task of automatic index assignment, as the definition of this attribute has completed the task of case representation.

In the current prototype, this index has been valued via template mining applied to the selected portions of substructures *abstract:main* and *body*. The incremental process of knowledge acquisition was used on 5% of the cases. This is the only attribute to which an alternative method was conceived, i.e., the reuse of cases with elaboration.

## 5. Retrieving and Reusing Experiences

Case retrieval results from the similarity assessment performed between each candidate case and the target case. Similarity assessment follows expert guidelines in terms of comparing and contrasting relevant values to the proper interpretation of the content of the legal experiences. The fact that similarity assessment is employed at the level of the values indicated by experts ensure a reliable comparison. The legal expert is the only one who knows what makes a legal experience similar to another. Following experts' guidelines ensures an effective similarity assessment.

Besides indicating at what dimensions to compare experiences, experts also specify how to compare them in terms of a range of similarity. For instance, if two values are absolutely different or similar, the values assigned are 0 and 1, respectively; if they are very similar is assigned a value .8.

The advantage of comparing legal experiences under expertise guidance is that it reduces the gap between usefulness and similarity. Useful experiences are more likely to be reused. Experts can indicate what types of values better index cases in order to explore their usefulness for further reuse. The representation of experiences through structured cases results in an abstract interpretation of the experience, another advantage in comparison to database approaches.

## 6. Textual CBR vs. Text Databases

The structured representation provided by the knowledge engineering effort results in an interpretation of the legal experiences proving another advantage besides enabling the similarity assessment between experiences. The valued features comprise an interpretation - an abstraction - that aims at providing to the expert the same information as the original text does. This results in a huge time savings, as experts and users do not need to read the whole textual description to understand and evaluate the usefulness of a given experience to the current problem.

Text databases employ statistical indexing that serves exclusively the purpose of retrieving cases. Text databases retrieve the whole texts from each query, forcing the users to read each text to decide their utility. The low precision of these systems causes the retrieval of many useless documents increasing difficulty of the task. Text databases evaluation of efficiency can be performed by two parameters: *recall* and *precision* Salton (1975). Recall is the proportion of useful documents that are actually retrieved from the base. Precision refers to the ratio of actually useful documents that are retrieved. Database systems have been found to be limited to a recall of 25% of relevant cases (Blair and Maron, 1985), meaning that the user has to read all the texts retrieved to conclude that only 25% are useful. The low quality of the retrieval may be dangerous in the legal domain where real relevant issues are under question.

The low accuracy of text databases might result from the use of statistical methods of indexing. Statistical methods do not use knowledge, i.e., they select terms depending upon their frequency of occurrence. By contrast, the similarity-based retrieval employed in CBR systems is essentially based on knowledge. A knowledge-based indexing process guarantees more efficiency, because precision increases as the indexes guiding similarity and retrieval are chosen with expertise knowledge increasing the chances of retrieving useful experiences. A statistical indexing might select two indexes that are two versions of the same expression, increasing the importance of the documents that use the two versions and decreasing the chances of an equally similar document that may have used only one version. Moreover, knowledge-based indexing avoids low levels of precision since the chances of retrieving useless experiences decrease.

## 7. Examples

A new session in PRUDENTIA begins with a research issue brought up by a judicial professional performing any legal activity. This legal expert becomes the expert user that uses PRUDENTIA to perform an intelligent jurisprudence research to figure out lessons and solutions to a given legal situation carried in mind.

The first goal of the system is to elicit this new situation from the expert user's mind. The user is first asked to fill out values for petition type and category and then asked to write down a brief summary of the situation that originated the session.

### 7.1 Example possession

In this example, the summary of the legal situation reads as follows: *"The defendant wants to appeal from his convictions of illegal possession of drugs based on the small quantity of cocaine confiscated that indicates the absence of commercial purposes".*

From the summary written by the user, the situation assessment process in PRUDENTIA is able to assign values for indexes as follows:

*petition type* = criminal appeal

| | |
|---|---|
| *category* = | illegal possession of drugs |
| *foundation* (1) = | commercial purpose |
| *foundation* (2) = | confiscated |
| *theme* (1) = | quantity of drug |
| *theme* (2) = | cocaine |

In the present example, the system considers the indexes assigned to be sufficient and uses these values to create the target case. The retrieval results in two cases valued with 100% similarity. The system offers an option where you can see all descriptors of the selected case. This is a very important feature because one of the complaints of users of available database systems is the necessity of reading the whole legal decision in order to identify the its usefulness. The way cases are modeled provides to the user the same result as a brief reading of the text.

If the user is motivated by a client who is researching for an appeal, one of the aspects to check is whether both cases scored with 100 had a positive result, meaning that the appeal was affirmed. Usually, a positive result is more likely to indicate a direction for the user who wants an affirmative result, while negative results can warn of possible failures. In this case, the first ranked case is the only positive, thus this is the natural choice of conducting research.

This example brings up the issue of whether the attribute *result* should be used as an index. A retrieval entailing *result* as an index would cause retrieval with higher scores in cases in compliance with the result desired in target case. These are two combinations of indexes that should be left to the user to decide.

*Besides the fact that the quantity of the drug is undersized, there are the lack of other elements to authorize the conviction in terms of the article 12 from the Law number 6.368/76, such as the identification of any witnesses that could have or had purchased the drug as well as devices usually used in drug traffic.*

Figure 2. Excerpt from legal decision text.

The view of the formlike representation of the selected case allows the user to understand and interpret the case without reading it. With such a knowledge-based interpretation of the experience, the user can decide whether or not to reuse such experience. In the current example, the user considers the context of the interpretation worthy of further reading.

From reading the legal decision, the user selects the excerpt laid out in Figure 2, since it teaches an important lesson. The excerpt in Figure 2 is a paragraph that states that besides the tiny quantity of drug, there are still other elements missing that are necessary to form a conviction in terms of the applicable law. The missing elements mentioned could be the identification of a witness who has actually acquired drugs from the defendant or even a device (such as a precision scale) used for commercial purposes.

The user should not stop the research process yet. There are two ways to continue: either searching for a second similar case with an affirmative result, or selecting visually by looking at the attributes of the first ranked retrieved cases.

Looking at the attributes that summarize the content of other cases, the user notes the fifth best ranked also has a positive result and has a value for foundation (2) = annulment. This suggests taking a closer look at other attributes and at the text. The interface showing the retrieved cases keeps a small window with part of the decision of the selected case. The fifth case in the rank, describes an explanation for an annulment, i.e., lack of consciousness of the intention to sell the drug. Taking a second look at the case attribute values, it is found that the value of *theme* (6) is violation of principle. From the decision, the user extracts another lesson to support argumentation, that is, "the violation of the correlation principle between sentencing and indictment nullifies decision".

The example demonstrates how easy it is for the judicial professional to perform effective jurisprudence research using PRUDENTIA. The example above also highlights the usefulness of keeping values for attributes that are lessons and not necessarily only indexes to guide retrieval. Let us now compare a similar search in a text database.

## 7.2 Comparison I

The first point of comparison refers to the selection of keywords to compose a query. The knowledge-based interpretation in PRUDENTIA provides a lexicon that allows words to be identified automatically by the system. That is, when the user types a small abstract of the situation, the system is able to recognize if there are words that are part in the lexicons. The closest procedure in a text database would be a manual search in the fields of words that occur in each field.

In the text database, the user has to compose a query. Let us build a query with the same words as the values assigned in the example in PRUDENTIA . The query is as shown in Figure 3:

"criminal appeal" cocaine confiscated commercial purpose illegal possession of drugs "quantity of drug"

Figure 3. Query for example "possession".

This query resulted 23 documents. The user now has 23 legal decisions to read in order to decide on their possible usefulness to the initial situation. According to Blair and Maron's evaluation, these 23 results indicate that there might be as many as 100 useful decisions in this base. We can agree with this estimate because:

- Possible misspellings have not been considered;
- Documents from other types and categories are not retrieved, since there is no partial matching;

•· Keywords are used at the same level of importance;

•· Mistakes are possible in building the query.

One solution to possible misspellings is the use of wild cards. However, even when available, wild cards delay the time of search significantly. The database system illustrated does not allow more than one wild card at the same query. Moreover, even if the system allowed as many wild cards as necessary, the user would have to look at the field of words manually, searching for every variation of occurrence of each word, a time-consuming job that few users would bear.

The query comprises an AND connector causing retrieval of only those documents that actually carry every word in it. This avoids the retrieval of any document that might have at least one alternative value for an index. According to expert interpretation, a document containing the same list of expressions with even two or three swaps might be also useful. An alternative would be the use of queries with other connectors such as OR, and XOR that are also available. The use of XOR is also exclusive and the connector OR would result in a cost-benefit paradox, since the more documents are allowed to be retrieved, the lower is the precision.

The problem is indeed the equivalence in the participation of indexes in the query. The expressions do not contribute equally in building a content and this is what prevents retrieval of a set of documents sorted by their relevancy.

Finally, we have to consider that even one wrong digit is enough to lower recall of a query. The chances of making mistakes increases with larger queries. The text database system showed in the example, for instance, does not have enough space to show intermediate results in large queries. We conclude that this system was not engineered with this purpose.

## 7.3 Example desertion

Another example refers to the legal situation that originates the research has the following basic indexes:

| | |
|---|---|
| *petition type* = | criminal appeal |
| *category* = | child desertion |
| *foundation* (1) = | good cause |
| *theme* (1) = | civil imprisonment |

Retrieved cases are laid out in Figure 1. The fifth column is indicates the category. The first seven cases are categorized with child desertion except for the fifth one. This emphasizes the question upon the reasons that might have caused this case to be retrieved. Even before viewing the attribute values or the legal decision, we conclude that this might be one case where the content is so similar that its importance grows in relation to the category. This is enough to assume that this case might be useful to the new situation.

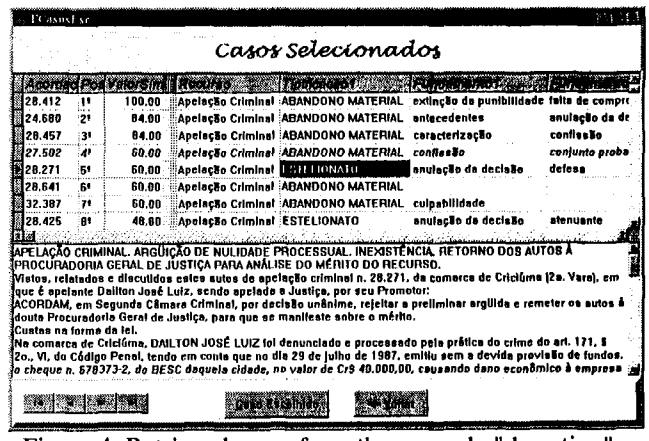

Figure 4. Retrieved cases from the example "desertion".

The fifth case ranked is labeled with number 28.271 in the first column (henceforth referred to as legal decision 28.271). This case is categorized as embezzlement. Reading the text we find that the decision affirms the defendant's appeal for nullity because the defendant had not been subpoenaed since he was already in custody due to a civil imprisonment. This is the lesson that justifies this case as a useful one to the research and demonstrates the case deserves the position among the best retrieved cases.

## 7.4 Comparison

Supposing that the same situation from the example *desertion* that originated this research had motivated a similar research in a text database system. The user has to build a query, and the same values that were assigned in PRUDENTIA are used as keywords, namely criminal appeal, child desertion, good cause, civil imprisonment.

There are two ways of using keywords in text databases: as primary indexes that partition the base; and as simple keywords, that search for every occurrence in the base. Values of the category, foundation, and theme may occur in any document associated to any type of petition. Hence, it seems that the type of petition should always be used as a primary index in order to reduce the possible number of documents retrieved. Conversely, if one tries to use the type of petition as a keyword, it will cause retrieval of every occurrence of this expression, even in documents of different types.

Using special designated fields as primary indexes always decreases precision in favor of a better recall. Therefore, to have a similar result for the research as the one obtained in PRUDENTIA, the search could not exclude other values of *category*, for instance, because that would avoid the retrieval of useful case such as the legal decision 28.271 described in the previous example.

## 8. Concluding Remarks

Initially, we have developed a prototype using only court decisions on habeas corpus petitions in murder lawsuits to demonstrate 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 in retrieval. The first prototype was developed in an application development tool and was tested with 22 cases.

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 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 the CBR problem areas such as similarity assessment and situation assessment. The implementation of the reasoner is essentially guided by expert domain knowledge.

The second prototype was then developed for two petition types in the criminal area: criminal appeals and habeas corpus. The case base comprises 138 cases that have been modeled semi-automatically.

The third prototype, PRUDENTIA Prototype III embodies a collection of 3,500 cases that have been autonomously converted into cases. The cases represent the experience of all criminal appeals that were submitted in the State Court of Santa Catarina in the period from 1990 to 1996.

The next stages will be to generate a case base, first for all habeas corpus petitions in this same period and next for all other petitions in the criminal area. This step will increase the size of the case base to approximately 10,000, and the knowledge already elicited will be reused. The required knowledge acquisition will focus on legal aspects related to the new categorizations of the remaining sub-domains. These next stages focus on improving execution time and precision.

The next big effort in knowledge acquisition takes place in the beginning of the modeling of decisions under the civil area. The inclusion of civil decisions in the case base comprises a new implementation of the methodology. As explained in section 5.2, the incremental process starts from reusing knowledge from the previous implementations and performs new knowledge acquisition processes while it is necessary. In the assignment of the foundation, most formal principles are reused. Concerning circumstantial values, new issues must be considered since there will be nothing such as murder weapon. However intention to cause harm is the same intention to commit a crime.

The knowledge embedded in texts has to be made available for future use. The problem of retrieving information and knowledge from texts stems from the increasing amount of information and knowledge that humankind must deal with. Storing such information and knowledge was facilitated by the advent of writing and it has become even easier with the computing technology. Hence, the problem of accessing information has only come up as larger amounts of information were stored. Retrieving the information demands a computational solution, but this solution should consider human needs and approaches to retrieving information and knowledge It seems appropriate to consider a solution that embeds the representation of some aspect of human cognition such as the analogical reasoning.

Textual CBR systems outperform text database systems in efficiency in retrieving knowledge and information from texts. The advantage of CBR systems stems mainly from the knowledge-based approach to indexing that is the essence of similarity-based retrieval. Since statistical methods select only indexes with a medium ratio of occurrence, terms that appear very often or very seldom may not be selected. In addition, terms with similar meanings might be selected misleading retrieval results. This seems to be the main reason of the low efficiency of such systems that supports the use of knowledge-based systems to the legal domain.

The use of a case-based reasoning system to retrieve textual documents is a means of representing the knowledge-based part of the search through AI technologies. In comparison with the use of a text database system, parts such as the construction of a query and the definition of the relative importance of the documents are automated. This increases efficiency since the search is not subject to human errors. Moreover, the intelligent system is consistent concerning domain knowledge.

## 9. References


Ashley, K. D. 1990. *Modeling Legal Argument: reasoning with cases and hypotheticals.* A Bradford book. The MIT Press, Cambridge, Massachussetts.

Blair, D. C. and Maron, M.E. 1985. An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. *Communications of the ACM* 28(3):280-299.

Blair, D. and Maron, M. A. 1990. Full-text information retrieval: further analysis and clarification. *Information Processing and Management* 26(3):437-447.

Blair, D. 1996. STAIRS Redux: Thoughts on the STAIRS Evaluation, Ten Years After. *Journal of the american society for information science* 47(1):4-22.

Branting, L. K. 1991. Building explanations from rules and structured cases. *International Journal of Man-Machine Studies* 34:797-837.

Branting, L. K. and Lester, J. C. 1996. Justification Structures for Document Reuse. In *Advances in Case-Based Reasoning: third European*

*Workshop; proceedings/ EWCBR-96*, 14-16, Lausanne, Switzerland, Ian Smith; Boi Faltings (ed.), Berlin: Springer.

Daniels, J. J. and Rissland, E. L. 1995. A Case-Based Approach to Intelligent Information Retrieval. In Proceedings of the SIGIR '95 Conference. Seattle WA, USA: ACM.

Gelbart, D. and Smith, J. C. 1997. FLEXICON: An Evaluation of a Statistical Ranking Model Adapted to Intelligent Legal Text Management. In the Fourth International Conference on Artificial Intelligence and Law, ICAIL-93, 142-151. Vrije Universiteit Amsterdam, The Netherlands: ACM.

Hahn, U. and Schnattinger, K. 1997. Deep Knowledge Discovery from Natural Language Texts. In Proceedings of the 3rd Conference on Knowledge Discovery and Data Mining, KDD`97.

Lawson, M.; Kemp, N.; Lynch, M. and Chowdhury, G. 1996. Automatic Extraction of citations from the text of english-language patents –an example of template mining. *Journal of information science* 22(6):423-436.

Lehnert, W. and Sundheim, B., 1991. A Performance Evaluation of Text-Analysis Technologies. *AI Magazine*:81-94.

Riloff, E. and Lehnert W., 1992 Classifying Texts using Relevancy Signatures. *Proceedings of the Tenth National Conference of Artificial Intelligence*, 329-334.

Riloff, E. and Lehnert W., 1994 Information Extraction as a Basis for High-Precision Text Classification. *ACM Transactions on Information Systems* 12(3):296-333.

Rissland, E.; Skalak, D. and Friedman, M. 1996. Using Heuristic Search to Retrieve Cases that Support Arguments. In *Case-Based Reasoning: Experiences, Lessons, and Future Directions*,111-124. David Leake, ed.:AAAI Press/The MIT Press.

Salton, G. 1975. *Dynamic Information and Library Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Smeaton, A.; Kelledy, F.; O'Donnell, R.; Quigley, I.; Richardson, R. and Townsend, E. 1995. Low Level Language Processing for Large Scale Information Retrieval: What Techniques Actually Work. In Proceedings of the Workshop on Terminology, Information Retrieval and Linguistics, CNR, Rome, October 1995. In Alan Smeaton's Online Publications. [Online] Available http://lorca.compapp.dcu.ie/~asmeaton/pubs-list.html, [September 12,1996].

Smeaton, A. 1997. Information Retrieval: Still Butting Heads with Natural Language Processing? In *Lecture Notes in Computer Science*, Pazienza (ed.):Springer-Verlag. In Alan Smeaton's Online Publications. [Online] Available http://lorca.compapp.dcu.ie/~asmeaton/pubs-list.html, [May 21,1997].

Smith, J.C. 1997. The Use of Lexicons in Information Retrieval in Legal Databases. In Proceedings of ICAIL-97, 29-38. Melbourne, Australia:ACM.

Smeaton, A. F. 1995. Low Level Language Processing for Large Scale Information Retrieval: What Techniques Actually Work. In Proceedings of a workshop Terminology, Information Retrieval and Linguistics:CNR, Rome.

Smith, J.C. and Deedman, C. 1987. The Application of Expert Systems Technology to Case-Based Law. In The First International Conference on Artificial Intelligence and Law, ICAIL-87, 84-93. Boston, MA:ACM.

Smith, J.C. 1997b (a work in progress) An Introduction to Artificial Intelligence and Law: or, Can Machines Be Made to Think Like Lawyers? In Artificial Intelligence and Law, The Focus of My Research and Writing by JC Smith. [Online] Available http://www.flair.law.ubc.ca/jcsmith/logos/noos/machine.html.

Smith, J.C., Gelbart, D. and Graham, D. 1992. Building Expert Systems in Case-Based Law. *Expert Systems with Applications* 4:335-342.

Sundheim, B.M. (ed.) 1992. *Proceedings of the Fourth Message Understanding Conference* (Defense Advanced Research Projects Agency. Los Altos, CA: Morgan Kaufmann.

Turtle H. R. 1991. Inference Networks for Document Retrieval. Ph.D. diss., Center for Intelligent Information Retrieval at the University of Massachusetts, Amherst. Available online http://ciir.cs.umass.edu/info/psfiles/irpubs/ir.html . [March 12, 1997].

Uyttendaele, C., Moens, M.F. and Dumortier, J. 1996. SALOMON: Automatic Abstracting of Legal Cases for Effective Access to Court Decisions. In JURIX.

Valente, A. 1995. *Legal Knowledge Engineering: A Modelling Approach*. (Amsterdam) and Omsha (Tokyo): IOS Press.

Weber-Lee, R.; Barcia, R.; Costa, M.; Rodrigues Filho, I.; Hoeschl, H. C.; Bueno, T.; Martins, A; and Pacheco, R. 1997a. A Large Case-Based Reasoner for Legal Cases. Lecture Notes in Artificial Intelligence: 2nd Int. Conference on CBR, ICCBR97. David Leake, Enric Plaza (ed.)-Berlin: Springer.

Weber, R. 1997b PRUDENTIA: Enabling a real world application of Case-Based Reasoning to jurisprudence research. Qualifying Examination approved by the Graduate Program of Production Engineering at the Federal University of Santa Catarina. In Activities. Available online http://www.eps.ufsc.br/~rosina/html/activities.html, [October, 1997].

Weber, R.; Barcia, R.; Pacheco, R.; Martins, A.; Hoeschl, H.; Bueno, T.; Costa, M.; and Rodrigues Filho, I. 1997c. Representing Cases From Texts In Case-Based Reasoning. III Congresso Internacional de Engenharia Industrial e XVII ENEGEP. Canela, RS, Brasil.

Weber, R.; Martins, A.; Mattos, E.; Bueno, T.; Hoeschl, H.; Pacheco, R.; von Wangenheim, C.; and Barcia, R. 1998. Reusing Cases to the Automatic Index Assignment from Textual Documents. In Proceedings of the 6th German Workshop On Case-Based Reasoning - Foundations, Systems, and Applications:249-255. Berlin.