2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

Computer Aided Legal Support Systems: Methodology to Automatically Convert the Legal Text into Legal Cases for Building Case Base Reasoning

Valliappan Raman [1]and Ayyappan Palanissamy [2]

[1] School of Computer Science, Swinburne University of Technology Sarawak, Kuching, Malaysia.
[2+] School of Business and Enterprise, Swinburne University of Technology Sarawak, Kuching, Malaysia

Abstract. In the legal domain, jurisprudence has an important role as a juridical source; its decisions support the application of the Law to a concrete case. In CALLS (Computer Aided Legal Support System), 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. The paper explains about Case representation for Case-Based Reasoning (CBR) that refers to selecting proper descriptors to describe and index cases. The complexity of case representation has been preventing CBR systems from solving problems when large case bases are required. In this paper we present the methodology for automatically convert legal texts into cases based on indexing methods and domain expert knowledge. The methodology is integrated to the domain of law although it can be extended to be applied to other domains as well.

Keywords: Legal Cases, Case Base, Indexing, Dimension.

1. Introduction

Judicial professionals have two sources of jurisprudence research: books and database systems. The search in the book is time-consuming and imprecise due to the limitations of humans' memory. Available text database systems do not guarantee the retrieval of useful documents. CALLS, the proposed case-based reasoner can be integrated to the current system which 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. Cases are units that describe an experience with dimension-value pairs (descriptors). Some of these descriptors guide retrieval and they are named indexes. Others describe lessons and solutions to solve the input problem. Identifying what dimensions better represent a case and which ones should be used for indexing comprehend the case representation problem in developing a CBR system. Representation of cases in CBR systems in which the knowledge is present in large corpus of texts is constrained by the subtask of text reading and interpretation to choose proper descriptors and the indexing vocabulary. Once we have figured out how to overcome these encumbrances, we can extend the application of CBR systems to several domains, comprising real world problems. Therefore, the breakthrough we require is a timely means of converting texts into a case-like representation that is an automatic approach to read texts and extract descriptors to describe and index the experience of these texts as cases. To achieve it, we proposed a methodology that uses an expert system that reads texts using domain expert knowledge assigns values to previously defined dimensions and extracts other dimensions from texts. This methodology has two steps: the first is its development and the second is its implementation.[+]

[2+] Corresponding author Tel: +6082416353; Fax: +6082423594; *Email:*apalanissamy@swinburne.edu.my

The main aim of the paper is to propose a CBR based the methodology for automatically convert legal texts into cases based on indexing methods and domain expert knowledge. The paper is organized as follows: In the next section, we give a background on the current research performed and in Section 3 concepts of AI and legal reasoning are mentioned, Section 5 explains the overview of CALLS-CBR system. Section 7 describes the methodology of converting legal text into cases in CALLS system is described .Section 8 explains the summary of how legal decision are mapped into cases in CALLS; finally the proposed work is concluded in Section 9.

2. Background

Strength of case-based legal reasoning has led to significant advances in modeling the way lawyers use previously decided cases. Some models generate legal arguments that to some degree simulate human adversarial discourse. The attempts in developing intelligent systems in the domain of law were boosted by HYPO [1] -- a CBR system that creates legal arguments from a case base on the domain of trade secret law. There have been different research works are carried on to retrieve legal text and cases. Hanney [2] demonstrate the systems in which the case base requires a complex modeling. The fact that the knowledge is embedded in databases does not mean that the case base is modeled. The indexing of a case base is oriented by the usefulness of a case when performing the reasoner task, which differs essentially from the indexing in information retrieval and database systems.

3. AI and Legal Reasoning

The law is an attractive domain for AI research for several reasons. First, the law has a tradition of examining its own reasoning process. Second, legal reasoning is stylized: one reasons according to stare decisis, with cases and by analogy. Third, much legal knowledge is readily accessible and relatively well structured, codified and indexed. Nevertheless it will not surprise, and may even please, lawyers to learn that the Restatement, the Uniform Commercial Code, and case law, like the theories of legal reasoning proposed by Karl Llewellyn and H. L. A. Hart, Rissland, are of limited immediate use to AI programmers [4][5][6]. Artificial Intelligence can act as: “A solution to the problem of legal complexity”. Computer programs can indeed solve legal problems. The fact that computer programs can model law is not necessarily simply of academic interest. Automated case research is one potential application of intelligent programs [9].

4. Overview of CALLS CBR SYSTEM

In this section we briefly explained the components and overview of CALLS-CBR system [10].

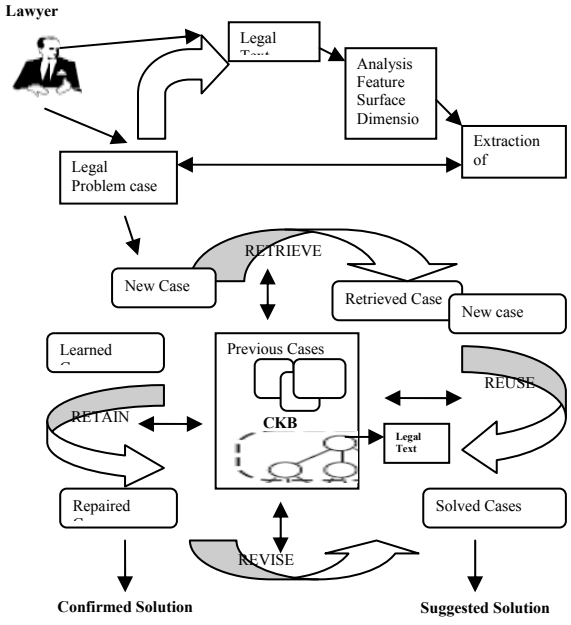

Fig 1 illustrates the CALLS –CBR system [10]

In brief, CALLS-CBR systems work as follows:

The cases that comprise the case memory describe legal texts. These texts are converted into cases automatically through the definition of attribute-value pairs that describe and index the cases [Fig 1].First, a problem case is input and analyzed to see what dimensions, sometimes also called factors, are applicable in the problem case. Dimensions address important legal aspects of cases and are used both to index and compare cases. They represent different argumentative approaches for dealing with an issue. Second, any case in the case-knowledge-base sharing at least one applicable dimension with the problem case is retrieved. These are considered the minimally relevant cases. Third, these relevant cases are sorted according to a model of on-pointness. In this sorting, which results in a partial order, Case A is considered more on-point than Case B if the set of applicable dimensions. A share with the problem case properly contains those shared by B and the problem case. Maximal cases in this ordering are called most on-point cases or mopc's. The result of sorting the cases can be shown in a so-called claim lattice. Those cases on the top level of the lattice are the mopc's. The problem case is the root node. Note, our CBR systems use the claim lattice as a starting point for various other aspects of CBR, such as the generation of arguments or creation of hypothetical.

5. Methodology: Converting Legal Text into Cases

In CALLS case base reasoning system, there are three levels to process the legal situation and provides the similar output cases, currently in this paper we have designed a case representation for representing problem cases and to build case knowledge base. The proposed methodology converts these texts into cases by implementing indexing methods. From the knowledge acquisition with domain experts, results the indexing vocabulary and the index assignment is performed automatically through rules. The legal cases of the Court are the description of petitions that comprehend parts of lawsuits. All these texts are written by judges who work for the Court and they have very similar backgrounds. Besides, there are some rules they are supposed to follow when writing these texts such as mentioning about the result of the petition in a paragraph right after describing what the parts are and which part applies for the petition. The main reasoning of the system is the search for the most similar legal cases. The reasoner's task is to retrieve the most useful cases to suggest solutions to the input problem. Within the domain of law, queries are made by judicial professionals who are able to understand and use the suggested lessons. The system we are discussing does not create arguments, but retrieve the most useful cases to help solving an input problem. It plays the role of a decision support system in the legal domain. The methodology comprehends text analysis, definition and assignment of fixed surface features and dimensions, and extraction of dimensions. These are all knowledge based steps and we are having two levels; firstly, the development level refers to knowledge acquisition to design the methodology; and, secondly, the implementation of the designed system to perform the steps developed. Currently our focus is on first level**.**

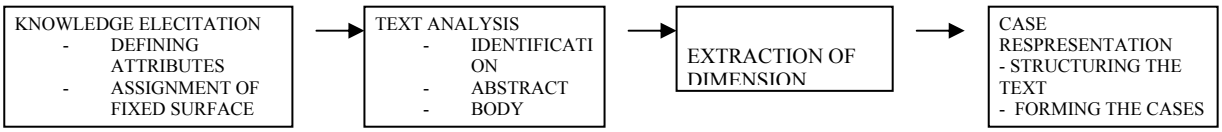

FIG 2 illustrates the Methodology of Converting Legal Text in to Legal Cases

7.1 Analyzing Text

The first module in CALLS is automatic conversion of legal text into legal cases, in that text analysis will be the first stage of development. The analysis of legal text aims on identifying the rhetorical structure of the documents. In the development of text analysis, experts indicate what structures to recognize in the texts and how to identify them. The final goal is to identify recognizable parts in the text that is where to find relevant information. Experts who write these legal cases follow some rules making this task easier. The development would require revision only if there is any change in the structure of the texts. The module receives the text as input and outputs the content of every structure and substructure that will be the input in the following steps of implementation. The implementation of text analysis is applied through rules in a logic programming module using Natural Language Processing.

Identification: surface features such as date, city, reporter and petition type.
.
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.
.
Body: 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. Conclusion: Court Decision
Closing: starts with one paragraph about votes followed by date, place and names of participating attornevs

Table 1 illustrates the sample rhetorical structure of the Legal Text

7.2 Fixed Surface Features and Dimension

The development of this second step started with the knowledge elicitation from domain experts who have defined a small set of fixed attributes to describe all legal texts [7]. Experts expect them to be valued in all cases. It is important to point out that the definition of the attributes does not require the experts to examine a significant amount of texts. Next, experts were asked to point the substructure where the value of each attribute is informed. 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 with NLP techniques in a module read the proper substructure and assigns values to the attributes. One of the fixed dimensions is category. The assignment of this value requires the use of NLP techniques because the category is not written clearly in a specified part of the text as the feature reporter is. However all possible values to be assigned to the dimension category are available in a "tree of categories", making the assignment easier. The expressions used to present the result may be refute, impugn, sanction or accept for one type of appeal whereas for another, the expressions used may be traverse, concede or disclaim. In this phase the rule-based system receives the texts and assigns values for all surface features and dimensions [9].

7.3 Extraction of Dimension

During the early stages of knowledge acquisition, experts pointed some expressions that indicate that there is either an illocutionary statement or a lesson [7]. This step refers to the modeling of rules that enable the system to automatically search for new dimensions using indicative expressions. Indicative expressions were pointed by the experts after an analysis of the samples of the text. Two examples of indicative expressions that the rule-based system searches for are the noun "impossibility" and the verb "certify". The experts provided knowledge with which the knowledge engineers could design heuristics to be deployed by the system in defining dimensions from each indicative expression found. A heuristic for the noun "impossibility" is based upon the idea that "impossibility indicates the condition of doing the action represented by the main verb in the sentence where impossibility is used. The implementation of this phase is employed by the module that searches for these expressions applying the proper heuristics whenever an expression is found. Based on the identified expressions, legal cases are represented.

6. Summary

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 form like 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. 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 CALLS 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 form like

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 iteration. 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 CALLS we make use of a form like 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 form like 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.

7. Conclusion and Future Work

In this paper, we have briefly surveyed how CALLS performs first stage of converting legal text in to legal cases for building the case base reasoner. The paper briefly explains the representation of cases, which made the expert's task possible for the development of a knowledge-based approach to build cases from texts. As a result of this methodology, we expect to overcome the difficulty in modeling cases that has been avoiding the launching of CBR systems in many real world applications, particularly where the knowledge available is in large corpus of texts. The paper focuses on two ideas: one is to search for lesson and indexing, second is to identify the structure of text and expressions in the text. Current work is in very early stage; only methodology is framed in this paper. In future, framed methodology and algorithm will be implemented, so after implementation it will be more useful for the legal domain application.

8. References

[1] Ashley, K. D., Modeling Legal Argument: Reasoning with Cases and Hypothetical's, Bradford Books/MIT Press: Cambridge, MA, 1990

[2] Hanney, Kathleen & Keane, Mark (1996). Learning Adaptation Rules From a Case-Base.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.

[3] Branting., L. Karl & Lester, James C, "Justification Structures for Document Reuse Advances in Case-Based Reasoning", third European Workshop, EWCBR-96, Lausanne, Switzerland, November 14-16, 1996.

[4] E. L. Rissland., J. J. Daniels, Z. B. Rubinstein, and D. B. Skalak," Case-Based Diagnostic Analysis in A Blackboard Architecture", In Proceedings, The 11th National Conference on Artificial Intelligence, pages 66–72,WashingtonD.C, July 1993.

[5] L. Thorne McCarty., Example Generation. In Third Biennial Conference of the Canadian Society for Computational Studies of Intelligence, University of Victoria, Victoria, B.C, 1980

[6] Hart H L A , Positivism and the Separation of Law and Morals. 71 Harvard Law Review 593, 1958.

[7] Kolodner, J, Case-Based Reasoning. Morgan Kaufmann, Los Altos, CA,1993.

[8] 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

[9] Aamodt, A. & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, MethodologicalVariations, and System Approaches. Artificial Intelligence Communications, 7 (1), 39-59.

[10] Raman Valliappan , Ayyapan Palanissamy, "COMPUTER AIDED LEGAL SUPPORT SYSTEM: An Initial Framework for Retrieving Legal Cases by Case Base Reasoning Approach", in IEEE 5th International Conference on Innovations in Information Technology, Dec 16-18 2008, United Arab Emirates.