Legal Knowledge Acquisition Using Case-Based Reasoning and Model Inference

Takahira YAMAGUTI and Masaki KUREMATSU
Department of Computer Science
Faculty of Enginnering
Shizuoka University
3-5-1, Jouhoku, Hamamatsu, Shizuoka, 432, JAPAN
e-mail:yamaguti@cs.shizuoka.ac.jp

Abstract

Although Case-Based Reasoning comes out in order to solve knowledge acquisition bottleneck, a case structure acquisition bottleneck has emerged, superseding it. Because we cannot decide an appropriate case structure in advance, a framework for CBR should be able to improve a case structure dynamically, collecting and analyzing cases. Here is discussed a new framework for knowledge acquisition using CBR and model inference. Model Inference tries to obtain new descriptors(predicates) with interaction of a domain expert, regarding the predicate as the slots that compose a case structure, with an eye to the function of theoretical term generation. The framework has two features: (1) CBR obtains a more suitable group of slots (a case structure) incrementally through cooperation with model inference, and (2) model inference with theoretical term capability discovers the rules which deal with a given task better. Furthermore, we evaluate the feasibility of the framework by implementing it to deal with law interpretation and certify two features with the framework.

1 Introduction

In the field of knowledge engineering, research on case based reasoning has been getting active in recent years, reflecting the difficulty in building an expert model in the development of expert systems. In CBR, however, this difficulty has not been resolved so much: the difficulty of building expert systems has been replaced with the difficulty of selecting appropriate case structures and effective retrieval and/or repair strategies based on them. That is, acase structure acquisition bottleneck has emerged, superseding the knowledge acquisition bottleneck. In the process of collecting and analyzing cases, a framework for CBR should be able to improve case structures dynamically.

On the other hand, in the field of model inference, there has been increasing interest in research on term generation for the first order language including the theoretical termgeneration issues[1]. Theoretical term generation means the operation to induce a useful concept and give it a name. It can be regarded as the operation to obtain new descriptors to define a problem.

From the above-mentioned background, a framework for knowledge acquisition, discussed here in this paper, starts as an attempt to obtain new useful descriptors through CIGOL[2]-based model inference by regarding the predicates as the slots that compose a case structure, with an eye to the function of theoretical term generation. The framework has the following features : (1) CBR obtains a more suitable group of slots (a case structure) incrementally through cooperation with model inference, and (2) model inference with theoretical term generation discovers the rules which deal with a given task better. Furthermore, we evaluate the feasibility of the framework by implementing it to deal with law interpretation and certify two features with the framework.

2 A Legal Knowledge Acquisition System

This section gives an outline of our framework and explains the knowledge acquisition system configurations of CBR and model inference with theoretical term capability.

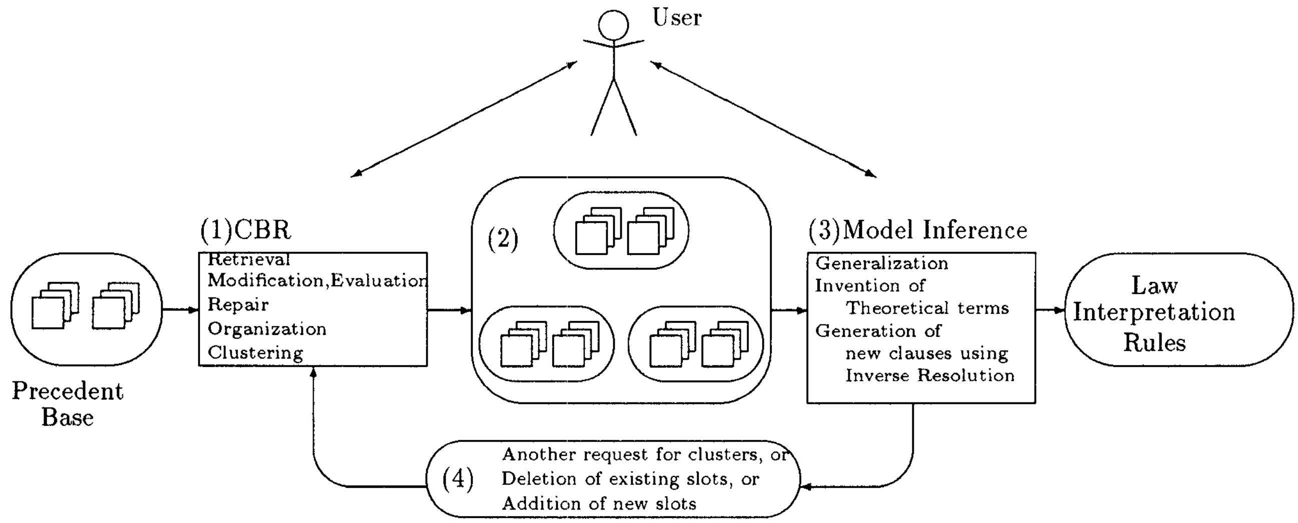

Figure 1. Cooperation between CBR & Model Inference in the Taskdomain of Law Interpretation

2.1 System overview

Figure 1 shows the framework for cooperation between CBR and model inference with theoretical term generation. The following explanation uses the numbers assigned in the figure.

(1) With the provision of a case base for each individual civil law precedent with a different case structure, CBR stores solved cases incrementally through retrieval, matching, modifying, evaluating, repairing, organizing and clustering.

(2) CBR gives a cluster to model inference. The cluster is a set of similar cases classified from the viewpoint of retrieval in CBR, and is converted to a set of ground clauses.

(3) With advice from a user, model inference tries to generalize the descriptor of the cluster (case description slots) and thereby invent new descriptors and compose relations among them as new clauses (rules). Consequently, some law interpretation rules can be obtained. If the model inference does not work well (e.g., it cannot invent a new descriptor), then the input is regarded as inappropriate. Then the model inference makes a request for input to CBR again and CBR gives a new cluster to the model inference.

(4) Besides the above request, the output from the model inference to CBR includes a group of unnecessary slots to be deleted and a group of new slot candidates to be added, giving an opportunity for case structure updating by CBR.

As explained above, the purpose of cooperation between CBR and model inference is to obtain rules to deal with a given problem successfully (law interpretation rules in this paper) by repeatedly updating the case structure and the analysis and synthesis of the descriptors to describe the case structure at each point of processing.

2.2 Configuration of CBR

(1) Initial case structure

The system has a group of general-purpose slots to describe the problem. Since each civil law has a specific slot group, the whole case structure results in putting these two types of slots group together, describing a case for which the applied civil law is known. With these slot groups, a priority slot group and non-priority one are ready at the retrieval phase. Figure 2 is an example of the case structure.

(2) Problem solving with CBR

Figure 3 shows an overview for the problem solving procedure with CBR[3].

In the retrieval phase, the system constructs a kind of claim lattice from HYPO developed by Ashley and others[4][5]. Here in this paper, a claim lattice tries to order relevant cases in terms of how on point they are to the root node, which is measured by the degree of overlap between a priority slot group of the root node and those of the retrieved case.

It retrieves the group of precedents which is most frequently identical with the value of the priority slot group with the problem case as the candidates for the best matched case, based on the claim lattice. Furthermore, from the candidates, the best matched case is the case which is closest to the value of the non-priority slot group of the problem case.

In the modification and evaluation phase, the best matched case is used to fill in the slot in the problem case whose value remains unknown. If the user does not accept the result after evaluation, then another best matched case is selected from the remaining candidates.

In the present stage of our system, it is assumed that a problem can be easily solved by repairing the modified case, the user considering on the content of the slot group shown by modified case. Nothing is therefore done in the repair phase.

In the organization phase, since the law applicable to the problem case has been decided, a claim lattice with the problem case as the root node is built based on the priority slot group contained in the general-purpose slot group and law specific one.

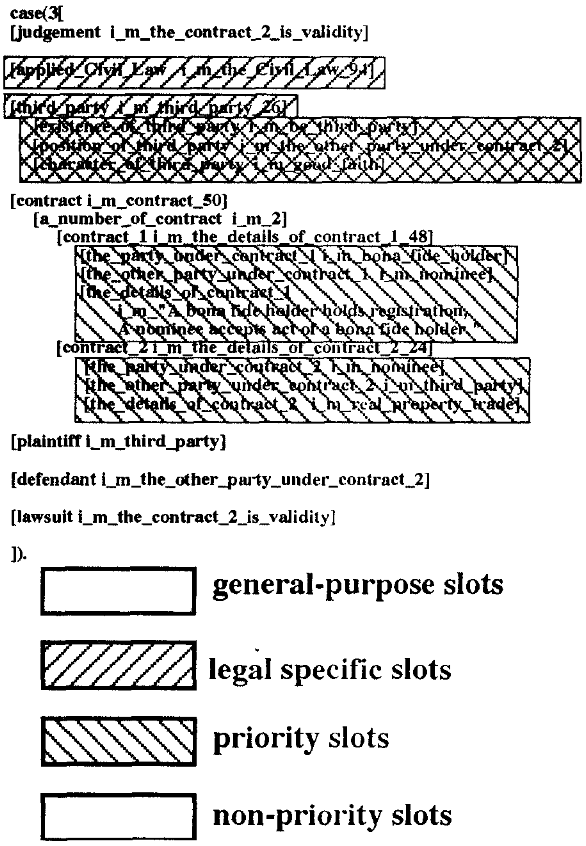

Figure 2. An Example of a Case Structure

(3) Cooperation with model inference

In the cluster generation phase, the root node and the child node (the candidates of best matched case) from the claim lattice are given as a set of ground unit clauses to the model inference.

After CBR would get the information about the addition of new slots and the deletion of unnecessary slots from the model inference, in the case structure transformation phase, the case structure could be updated, the user filling the value of an unknown new slot in. Since the information about slots would change the priority slot group, a new claim lattice might be re-organized.

In this manner, through cooperation between CBR and the model inference, the case structure and the retrieval strategy could be improved and the performance of CBR would be expected to be enhanced through the improvement of the priority slot group.

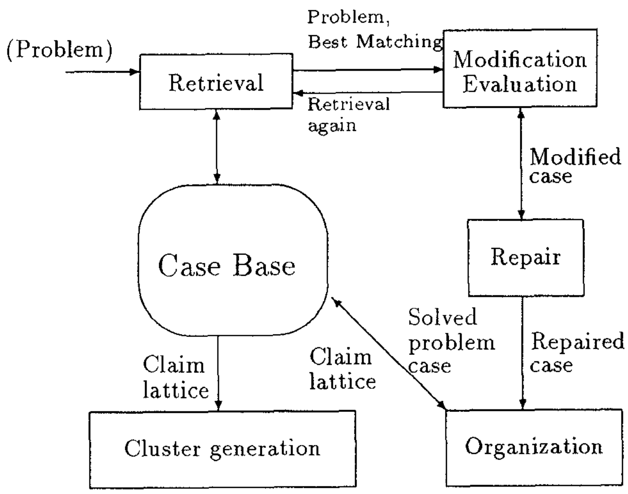

Figure 3. A Framework for CBR

2.3 Configuration of model inference

The model inference mechanism is an extension version of the CIGOL.

CIGOL is a system which applies three types of inverse resolution operators, namely, Truncation, V-op(erator) and W-op(erator), to a set of ground input unit clauses and thereby outputs a set of general clauses which the set of input clauses are derived from. Assuming that two given clauses are one parent clause and a resolvent, V-op attempts to find another parent clause which should have been given as a theorem in advance, as shown in Figure 4. W-op is an operator to realize theoretical term generation. It attempts to cut out a concept under the restriction that a new concept (predicate) is subordinate to the existing concept, as shown in Figure 5. The user gives it a name. Truncation can be regarded as a special case of W-op. Actually, it tries to generalize within the scope of the existing clause structure, as shown in Figure 6. Since, however, Truncation and V-op involve some nondeterminism at the level of generalization, the user must decide an appropriate level of generalization.

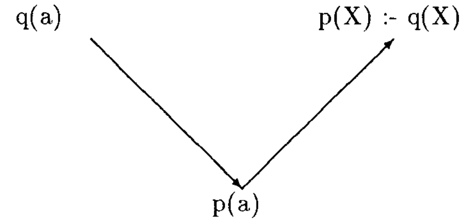

Figure 4. V-operator

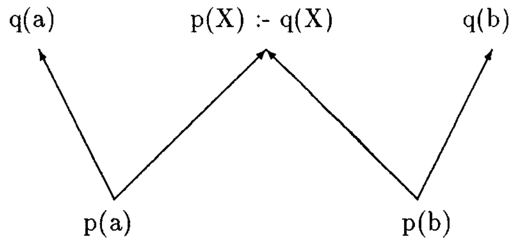

Figure 5. W-operator

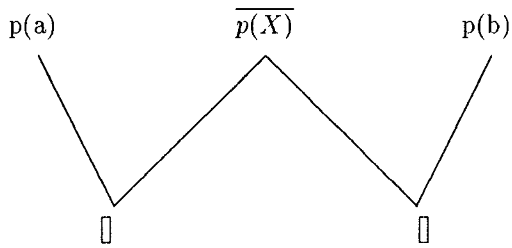

Figure 6. Truncation

Execution control through the model inference is done in the following manner: It calls Truncation and then controls the execution of V-op and W-op by means of the size value that represents the degree of specialization of the clause. Priority is given to the operator with a smaller size value.

The size of syntactic objects is defined as follows:

- size-of clause set $\{C_1, \ldots, C_n\} = \sum(\text{size-of clause } C_i)$ $\{1 \le i \le n\}$
- size-of clause $\{L_1, \ldots, L_n\} = \sum(\text{size-of literal } L_i)$ $\{1 \le i \le n\}$
- size-of literal or term $\{f(t_1, \ldots, t_n)\} = 2 + \sum(\text{size-of } t_i)$ $\{1 \le i \le n\}$
- size-of variable : v=1
- size-of constant : c=2

The process of CIGOL can be regarded as one of acquiring a set of general non-unit clauses from a set of ground clauses, finding subordinate concepts to a given concept as shown in Figure 7.

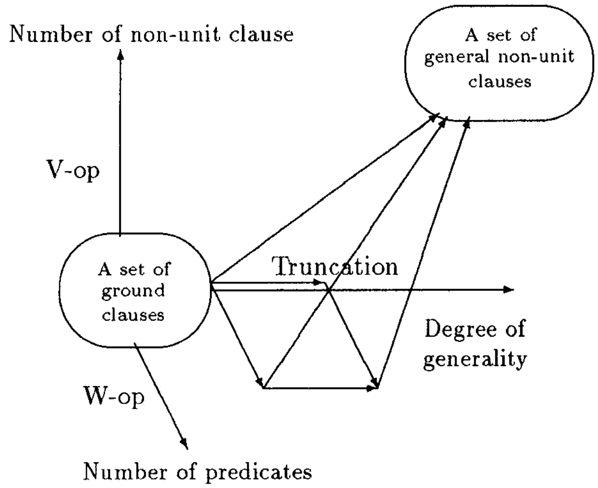

Figure 7. Significance of CIGOL

This framework has a feedback from model inference to CBR. W-op and Truncation of CIGOL are to instruct CBR to add new lower slots to the existing slots and to delete unnecessary slots, respectively.

The following are the enhanced CIGOL functions to activate cooperation between the model inference and CBR:

(1) Enhanced Truncation

Truncation is executed just for a set of unit clause which have the same predicate. To expand the scope of search for unnecessary slots, it has been done to a set of non-unit clauses which have the same clause structure.

(2) Enhanced W-op

W-op is only to derive a single predicate. It has been improved so that multiple predicates can be derived, leaving the combination of predicate arguments to the user. At this time, messages given to the user have been made easy to understand with the information about the types of arguments of predicates.

(3) Enhanced V-op

V-op attempts to perform inverse resolution for all pairs of clauses as far as possible, generating so many unnecessary clauses. To eliminate unnecessary clauses similar to the inversely resolved ones rejected by the user, the following elimination strategy has come into V-op:

Elimination strategy

V-op eliminates any pair of predicates that holds the implication rejected by the user.

3 Applications to legal interpretation

It is so difficult for a novice to apply the appropriate law to a given problem, because the law is too general to apply to it. To fill the gap, threrfore, lawyers have law interpretation rules from past precedents and legal theory.

By way of an example, this section takes up the problem of applying analogy that is often discussed when applying Article 94 of Civil Law, which relates to the declaration of intention. According to T. Tsubaki[6], the application of analogy is defined as an attempt to interpret an event not directly stated in laws or ordinances, using some appropriate rules derived from precedents and theories. If, therefore, this framework would discover law interpretation rules including descriptors invented from precedents and theories, it could be evaluated as rules that be used in the application of analogy.

3.1 the Article 94 of Civil Law

An outline of article 94 of civil law is as follow:

1) A fictitious declaration of intention made by a declarant in collusion with the other party is invalid.

2) But the declarant or the other party can not claim that the declaration of intention is invalid, in the case of a bona fide third party does not know that the declaration of intention is invalid.

3.2 System evaluation

The new effective predicates of *fictitious_appearance/1* and *act_ot_the_other_party_under_fictitious_appearance/1* have been introduced by W-op and contributed to build up an effective law interpretation rule related to Article 94 of Civil Law. Furthermore, they have been returned CBR in order to update the case structure, resulting in the retrieval efficiency of CBR being improved, as shown in Figure 8. This means that they have been useful to identify the places in the general-purpose case structure where additional information is to be collected.

The model inference has generated two clauses, as shown in Figure 9.

One clause generated by W-op consists of predicates, which came from law, and new invented predicates. Their new predicates are regard as legal requisites for applying analogy. Therefor, this clause is regard as a useful law interpretation rule related to Article 94 of Civil Law.

The other clause generated by V-op is regard as the theory that regards the lower limit of implicit approval as mere leaving. The clause has the same meaning as the hypothesis in the reference[6].

It can therefore be said that the system has successfully composed useful law interpretation rules.

We have already discussed the strategy to eliminate unnecessary clauses obtained through the application of V-op. It uses the information about any pair of predicates that should not hold implication relationship. As a formal strategy to eliminate them, it is possible to use the size value used for execution control. Specifically, any inversely resolved clause with a size larger than a specified upper limit size value is regarded as unnecessary due to excessive specialization.

Table 1 shows us the result of examining to what extent unnecessary clauses can be eliminated through the combination of the first strategy (setting an upper limit to size values) and the second strategy (using the information about any pair of predicates that should not hold implication relationship):

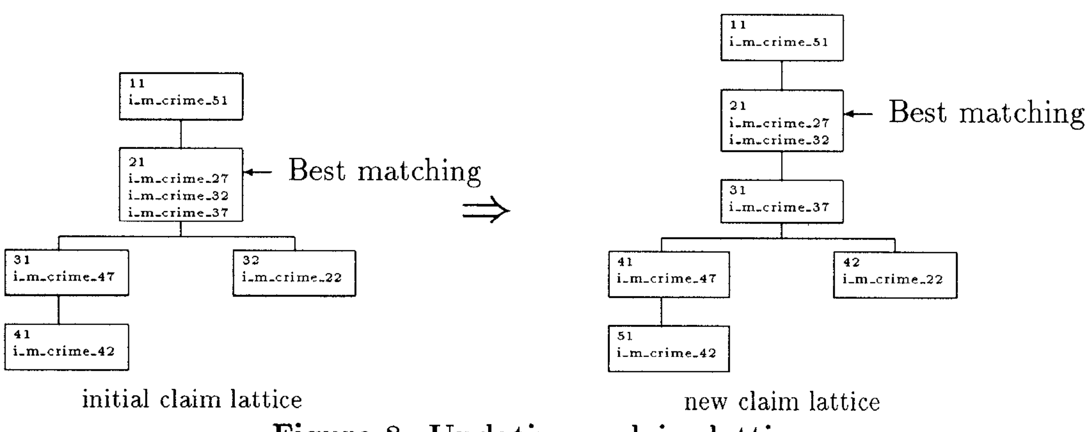

Figure 8. Updating a claim lattice

[1] judgement(the_contract_2_is_validity):-
contract_1(the_party_under_contract_1(bona_fide_holder), the_other_party_under_contract_1(nominee), the_details_of_contract_1(A,B)),
contract_2(the_party_under_contract_2(nominee), the_other_party_under_contract_2(third_party), the_details_of_contract_2(sale(real_property),buy(real_property))),
third_party(existence_of_third_party(be_third_party), position_of_third_party(the_other_party_under_contract), character_of_third_party(good_faith)),
plaintiff(third_party), defendant(the_other_party_under_contract_2), lawsuit(the_contract_2_is_validity),
simulated_appearance(A),
act_of_the_other_party_under_simulated_appearance(B).
[2] act_of_the_other_party_under_simulated_appearance(acceptance(A)):-
act_of_the_other_party_under_simulated_appearance(leave(A)).

Figure 9. Generation of Law Interpretation Rules

Table 1 : A Number of Questions and Rules Using V-op.

Questions with the First Strategy	Size Value	Questions with Both Strategies	Effective Rules
0	-10	0	0
0	-5	0	0
9	0	1	0
23	5	8	1
37	10	13	1

From Table 1, when the upper size value is larger than five, the model inference system finds an effective law interpretation rule through the combination of the first strategy and the second one. However, because it is so difficult to set an appropriate size value in advance, we will have to consider refined strategies for eliminating unnecessary clauses generated in the model inference.

4 Conclusion

By using a law interpretation problem by way of an example, we have proposed a framework for knowledge acquisition through cooperation between CBR and model inference. The framework has the following features: (1)CBR obtains more suitable group of slots (a case structure) incrementally through cooperation with model inference, and (2)model inference with theoretical term generation discovers the rules which deal with a given task better. Now we are working to refine the cooperative process between CBR and model inference and to eliminate unnecessary clauses using another strategy in model inference.

Acknowledgments

We would like to express our thanks to Professor Suzuki and Mr. Adachi who have discussion about the design of CBR together, and Professor K. Yoshino and the members of legal expert systems study group.

References

[1] S.Muggleton editor : Inductive Logic Programming, Academic Press(1992)

[2] S.Muggleton, W.Buntine : Machine Invention of First-order Predicates by Inverting Resolution, Proc. 5th ICML, pp.339–352 (1988)

[3] C.K.Riesbeck and R.C.Schank : Inside Case-based Reasoning, Lawrence Erlbaum Associates, Inc (1989)

[4] K.D.Ashley and E.L.Rissland : A Case-Based Approach to Modeling Legal Expertise, IEEE Expert, pp. 70-77 (1988)

[5] K.D.Ashley : Toward a Computational Theory of Arguing with Precedents: According Multiple Interpretations of Cases, Proc. 2nd Inter. Conf. on Ai and Law, pp.93-102 (1989)

[6] H.Tubaki : Application of analogy for the Civil Law, Journal for Law Vol62, No7, pp.72-77 (1987)