Thirteenth Australian Computer Science Conference

ACSC-13

7-9 February, 1990

Australian Computer Science Communications
Volume 12, Number 1

Department of Computer Science
Monash University
Clayton, Vic
AUSTRALIA 3168

Legal Expert Systems: The Inadequacy of a Rule-based Approach

James Popple

Department of Computer Science
Australian National University

Abstract

The two different categories of legal AI system are described, and *legal analysis systems* are chosen as objects of study. So-called *judgment machines* are discussed, but it is decided that research in legal AI systems would be best carried-out in the area of *legal expert systems*. The process of legal reasoning is briefly examined, and two different methods of legal knowledge representation are discussed (rule-based systems and case-based systems).

It is argued that a rule-based approach to legal expert systems is inappropriate given the requirements of lawyers and the nature of legal reasoning about cases. A new approach is described, incorporating both rule-based *and* case-based knowledge representation. It is claimed that such an approach can form the basis of an effective and useful legal expert system.

CR Categories and Subject Descriptors: **I.2.1** [Artificial Intelligence]: Applications and Expert Systems—*law*; **I.2.3** [Artificial Intelligence]: Deduction and Theorem Proving—*answer/reason extraction, deduction, logic programming, uncertainty, "fuzzy" and probabilistic reasoning*; **I.2.4** [Artificial Intelligence]: Knowledge Representation Formalisms and Methods—*representations (procedural and rule-based)*; **I.2.5** [Artificial Intelligence]: Programming Languages and Software—*expert system tools and techniques*; **I.2.6** [Artificial Intelligence]: Learning—*analogies, induction*; **J.1** [Computer Applications]: Administrative Data Processing—*law.*

Key Words and Phrases: case-based systems, expert systems, law, legal reasoning, rule-based systems.

1 Introduction

Legal AI systems can be divided into two categories:

- legal retrieval systems; and
- legal analysis systems.

Legal retrieval systems allow lawyers to search through databases, containing details of statutes and decided cases, for information. AI techniques may be employed to simplify this task (*e.g.* by searching for keywords which have not

been input by the user but are deduced to be equivalent to, or sufficiently related to, the input keywords).

Legal analysis systems take a set of facts and determine the ramifications of those facts in a given area of law.

(McCarty (1980a) identifies a third category of legal AI systems: integrated legal systems. He cites as an example computerized title registration systems which make decisions about people's rights and obligations.[1] It is hard to see why such a system could not be usefully classified as a legal analysis system, albeit with some of the features of a legal retrieval system.)

Mehl (1959) claims that there is no fundamental difference between these two categories (legal retrieval systems and legal analysis systems)—that the difference is one of degree only.[2] But Shannon and Golshani (1988) point out that the difference between a system based on a "conceptual model of legal analysis" and text-retrieval systems is that the latter do not "understand" any area of the law.[3]

This paper will be concerned with legal analysis systems.

2 Legal Analysis Systems

Legal analysis systems can be divided into two categories:

- *judgment machines*: systems that make a judge-like pronouncement (*e.g.* "X is guilty of offence Y for the following reasons ..."); and
- *legal expert systems*: systems that provide advice similar in form to that which a solicitor might provide (*e.g.* "The facts in this case are similar to those in P *v* D where the defendant was found guilty, but the instant case can be distinguished from P *v* D as follows ...").

2.1 Judgment Machines

The idea of a *judgment machine* was raised thirty years ago by Mehl (1959). Although such a machine would perform the functions of a judge, it was said that a role for humans remained because:

> ... the solution to a legal problem may depend upon extra-rational factors, involving the whole of human experience ...[4]

Almost twenty years later, D'Amato (1977) suggested that a judgment machine could *replace* a human judge. His proposed machine would take the relevant facts as input and produce a number in the range -1.0–1.0 (where a positive number indicates a victory for the plaintiff). Given the multiplicity of factors, he claimed, a result of zero would be extremely unlikely.[5] Somewhat begrudgingly,

---

[1] McCarty (1980a), p. 2.
[2] Mehl (1959), p. 759.
[3] Shannon and Golshani (1988), p. 306.
[4] Mehl (1959), p. 758.
[5] D'Amato (1977), p. 1280.

he allowed for some vestige of human control. An appeal court could review all of the machine's determinations in a certain numerical range (*e.g.* -0.05–0.05) within which the cases would be so close that a re-examination might be required. The review court's subsequent decision would then be incorporated into the system.[6]

The idea of human judges being replaced by machines has been vehemently criticized. According to Weizenbaum (1976):

> The very asking of the question, "What does a judge ... know that we cannot tell a computer?" is a monstrous obscenity. That it has to be put into print at all, even for the purpose of exposing its morbidity, is a sign of the madness of our times.
>
> Computers can make judicial decisions, computers can make psychiatric judgments. They can flip coins in much more sophisticated ways than can the most patient human being. The point is that they *ought* not be given such tasks. They may even be able to arrive at "correct" decisions in some cases—but always and necessarily on bases no human being should be willing to accept.
>
> ... What emerges as the most elementary insight is that, since we do not now have any ways of making computers wise, we ought not now to give computers tasks that demand wisdom.[7]

Moles (1987) agrees:

> The computer scientists, encouraged by the modern positivists, fail to recognize ... that law, positive morality and ethics are inseparably connected parts of a vast organic whole. Judgments are involved at every stage of the legal process and machines cannot make judgments. In stating that legal rules can be applied without further judgment; that they apply in an all or nothing fashion; that legal decision making follows the form of the syllogism or that it is a pattern-matching routine, the modern positivists, joined now by the computer scientists take us along a dangerous road.[8]

But D'Amato sees advantages in replacing human judges by machines:

> Would we lose a judge's "judgment," and how important would such a loss be to our legal system? Surely computers do not make "judgments" the way humans do, and so we would lose the "human" aspect of legal judgments. But what specifically do we lose when we lose the humanness of judgments? Is human judgment just a euphemism for arbitrariness, discretion, or bias?[9]

---

[6] D'Amato (1977), pp. 1290–1291.
[7] Weizenbaum (1976), pp. 226–227.
[8] Moles (1987), p. 271.
[9] D'Amato (1977), p. 1281.

Proponents of the idea of automated judges claim that such systems would reduce the cost of the legal system, find inconsistencies in the law, and provide a level of certainty in the law which does not exist at present.[10]

In 1977, D'Amato claimed that his proposal for a computerized judge was a modest one.[11] Yet, the current state of AI technology is such that no judgment machine has been implemented which can pass judgment in any substantial area of law. The ethical question of whether judges *ought* to be replaced by machines remains a hypothetical one. The author's sympathies lie with Weizenbaum and Moles. For that reason, and because AI technology has not yet caught up with hypothetical judgment machines—machines which may prove impossible to build—this paper will, henceforth, be concerned only with legal expert systems.

2.2 Legal Expert Systems

For the purposes of this paper, a *legal expert system* (LES) will be defined as a system that provides answers to legal questions which resemble the answers one might expect from a lawyer.

This definition excludes AI systems which might merely assist a lawyer in coming to legal conclusions or preparing legal argument (*e.g.* a sophisticated legal retrieval system). This is not to say that a lawyer should not be able to use an LES, just that the output from an LES should not require further legal analysis. This output should be in such a form that it can be the basis of a lawyer's legal argument in court.

LESs are not judgment machines; they will not usurp judicial power. The development of sophisticated LESs will not remove the need for lawyers, although it may change the nature of some legal work. An LES should be a powerful tool for use by both lawyers and non-lawyers.

3 Legal Reasoning

All LESs must be capable of legal reasoning, or (at least) of simulating legal reasoning. According to Shannon and Golshani (1988), legal reasoning (with a statute and its related case law) requires *at a minimum* that the lawyer:

1. explores the facts, and determines the type of case and which statute applies;

2. examines the words of the statute

   > ... hoping to extract meaning from the morass.[12]

   and formulates that meaning into rules;

---

[10]This increased level of certainty would be a result of freely available (automated) judicial advisory opinions.

[11]D'Amato (1977), p. 1288.

[12]Shannon and Golshani (1988), p. 307.

3. matches the facts of the case against the statute and attempts to reach some conclusions; and

4. (possibly) seeks additional information as to the meaning of the statute, and the facts of the case.[13]

Elements of this reasoning process are necessarily reflected in the design of an LES.

According to McCarty (1980a), an LES must be able to represent:

- the "facts", at some comfortable level of abstraction; and
- the "law", which would consist of a system of "concepts" and "rules" which:
  - would be relatively abstract (*i.e.* they would subsume large classes of lower-level factual descriptions); and
  - would have normative implications (*i.e.* they would specify which actions were permitted and which were obligatory).[14]

McCarty's TAXMAN project was concerned with the area of corporate tax law. The basic "facts" of a corporate case were captured in a relatively straightforward representation (*e.g.* corporations issue securities). Below this level was an expanded representation of the meaning of various things (*e.g.* a security interest) in terms of their component rights and obligations. Above this level—presumably above *both* levels, although this is not made clear—was the "law" (statutory rules which classify transactions as taxable or non-taxable *etc.*).[15] Legal analysis, according to McCarty, is a simple matter of applying the "law" to the "facts".

McCarty's approach is attacked by Moles (1987):

> McCarty appears not to appreciate that 'corporations', 'securities', 'property', 'dividends' and so on are not subsumed 'beneath the law', but are each the products of complex legal analysis. The question of whether certain transactions are taxable or not is intimately tied into that legal analysis.[16]
>
> The sad thing is that he has not shown the slightest awareness of the nature of the legal enterprise.[17]

Moles is opposed to the very idea of an LES, but his attack on McCarty's approach is important because it emphasizes the complexity of the task of building an LES. McCarty accepts that some legal concepts have what is called an *open texture*, which makes them very difficult to represent. Moles argues that *all* legal concepts are open-textured—that even those concepts which McCarty claims are easily represented are impossibly complex (in the sense that they are beyond the capability of machines).

---

[13]Shannon and Golshani (1988), pp. 306–307.
[14]McCarty (1980a), p. 3.
[15]McCarty (1980a), pp. 4–5.
[16]Moles (1987), p. 270.
[17]Moles (1987), p. 269.

3.1 Open Texture

The term *open texture* was first used in jurisprudence by Hart (1961). According to Shannon and Golshani (1988):

> Roughly speaking, a concept is open-textured if it defies complete definition.[18]
>
> Here we refer to the inherent indeterminacy of the meaning of words that are used to describe the predicates of statutes. Sometimes other factors, such as vagueness, may also be considered as part of the open texture issue.[19]

The problem of open texture in legislation is one which arises in the development of any LES. A naïve solution to the problem is to examine the cases construing the relevant section of the statute and produce new rules which help to define the meaning of the open-textured concept. Of course, the new rules which are produced in order to help to define the open-textured concept will probably contain new predicates which are also open-textured. As discussed in §4.2.1, this approach has severe limitations.

4 Knowledge Representation

The method of representing legal knowledge in an LES depends upon the nature of the area of law with which the LES is concerned: *statute law* or *case law.*

4.1 Statute Law

A number of projects have focussed on representing the provisions of a statute as a set of rules. When these rules are applied to the facts of a case (*i.e.* by instantiating previously free variables) an inference engine can produce an answer which represents the effect of the statute on the given facts. For example, the *British Nationality Act* has been encoded as a PROLOG program (Sergot *et al.*, 1986).

Shannon and Golshani (1988) claim that extracting the rules from a statute in an *ad hoc* fashion is unsatisfactory because:

- subsequent designers/users cannot trace the evolution of a set of rules from the words of the statute;
- the rules cannot be mechanically checked for correctness; and
- this *ad hoc* approach may lead to rule formulations which do not work well together.[20]

---

[18]Shannon and Golshani (1988), p. 312.
[19]Shannon and Golshani (1988), p. 312, n. 10.
[20]Shannon and Golshani (1988), p. 308.

It is admitted that it is not possible to check such rules for correctness, but it must be remembered that a lawyer is similarly unable to check her/his own interpretation of a statute for correctness. It is up to the knowledge engineer to check that the rules are an accurate representation of the statute. Shannon and Golshani suggest that it is unlikely that a mechanical method can be developed for transforming statutory language into formal rules, but methods have been developed which (if followed rigorously) reduce the likelihood of error.[21] The use of such methods would also reduce the scope for dissimilar rule formulations. The first problem (being unable to trace the evolution of the statute to the rules) can be obviated by the sensible use of comments.

4.2 Case Law

No area of law is covered exclusively by statute.[22] Case law is fundamental to the Australian legal system, which relies heavily upon the *doctrine of precedent*. This doctrine states that each decided case is not merely an example that later judges may chose to follow, or to ignore: that case, itself, becomes part of the law. This means that any useful LES must take account of the law embodied in previously-decided cases (the *common law*).

The problem of representing case law is more complex than that of representing statutory provisions. This problem is related to the problem of open texture (discussed in §3.1). When the meaning of a statutory provision is unclear, the courts give meaning to that provision. When faced with this problem the LES builder has two options:

- to pose the question to the user ("What is the meaning of this open-textured concept?"), and to accept the user's answer as an accurate statement of the law; or
- to examine the cases and incorporate, in the knowledge base, the meaning that those cases ascribe to the open-textured concept.

The first option is satisfactory only if the LES is being used by a legal expert who is (presumably) in a position to answer the question. This approach would surely reduce the LES's usefulness. Shannon and Golshani (1988) opt for a combination of both approaches:

> This model allows some room for reasoning with the facts but relies on the user for input when no clear inference is found. We add depth to our model by filling in the basic rules and definitions of the statute with additional factual examples from decided cases.[23]

Tyree *et al.* (1988), with their FINDER system, take a completely different approach to the problem posed by case law. The area of law which they chose

---

[21] Shannon and Golshani (1988), pp. 308–309.

[22] Even a new statute, which has not specifically been the subject of any case, is interpreted in the light of previously-decided cases.

[23] Shannon and Golshani (1988), p. 311.

to model is based entirely on case law. This makes it inappropriate to use a rule-based system. According to Tyree *et al.*:

> It is not that it is theoretically impossible to write such rules, but that it is not the natural way in which lawyers reason with cases.[24]

Further, they claim, the number of decided cases in a given area of law is usually so small that inductive tree generation algorithms cannot be used.[25]

4.2.1 Rule-based Systems and Case Law

In fact, it is *not* possible to formulate production rules which will adequately represent case law, because such an extended rule-based system would be of little use to a lawyer. This defect in rule-based LESs goes beyond the simple fact that rule-based reasoning "it is not the natural way in which lawyers reason with cases."[26] Such a system may be capable of producing an answer (possibly with an attached estimate of its probability). But a lawyer is not interested in a simple answer, even if it is strongly suggested by a long line of legal authority.

A lawyer argues about the meaning of an open-textured concept by reference to previously-decided cases. No two cases can be *completely* identical, given the plethora of facts associated with any given case. Of course some of these differences may be insignificant, and much of a lawyer's reasoning by analogy concerns the *legal* significance of these differences. Roughly speaking, a lawyer argues with cases in the following fashion:

- If the result of a previously-decided case is desirable, she/he argues that there are no legally significant differences between the previous case and the instant case, so the previous case should be followed.

- If the result of a previously-decided case is undesirable, she/he argues that there is some legally significant difference between the previous case and the instant case upon which the previous case should be distinguished.[27]

No amount of reason extraction from an inductive rule-base will provide the information that a lawyer needs to argue in this fashion. So, it can be seen that attempting to reduce the results of previously-decided cases to rules which can be simply added to a statutory rule base is an inappropriate approach to the problem of open-textured concepts. As Tyree *et al.* state, such an approach does not reflect the way in which lawyers reason about cases. But, more importantly, it makes for an inadequate LES.

---

[24]Tyree *et al.* (1988), p. 232.

[25]Tyree *et al.* (1988), p. 232.

[26]If an LES (or any decision support system, for that matter) can produce useful answers, it is of little relevance whether the reasoning it employs to produce those answers accurately models a lawyer's reasoning (or the reasoning of the appropriate domain expert).

[27]This process is, of course, greatly complicated by the different importance which is placed upon decisions of different courts.

4.2.2 The FINDER System

The FINDER system of Tyree *et al.* (1988) takes the following approach to cases. Expert knowledge is used to determine the most important cases in a given (fairly small) area of law, and the attributes which are of legal importance to the outcome of those cases. These attributes are given weights—not by a legal expert, but by examining the extent to which each attribute differs across the cases. Using these weighted attributes it is possible to measure statistical *nearness* (similarity) between the cases.

When the facts of the instant case (*i.e.* those attributes which are of legal importance) are entered, the nearest previously-decided case (the *nearest neighbour*) is ascertained. If the attributes of the nearest neighbour are the same as those of the instant case then the answer is clear. When the attributes of the cases differ, the system gives details of the nearest neighbour, and lists the differences. To avoid giving misleading advice, the system also finds the nearest case which reached the opposite conclusion to that of the nearest neighbour (the *nearest other*).[28] That case, and the differences between it and the instant case, are explained. To further reduce the chance of mistakes, several statistical techniques are employed to ensure that the nearest case is not greatly different from the instant case.

5 Suggestions for Future Legal Expert System Research

As has been shown, an LES represents its legal knowledge in one of two ways. Statutory law is represented by rules. Case law can also be represented by rules, but this approach is problematic. The FINDER system stores a collection of attributes for each of the relevant cases. By comparing these attributes with those of the instant case, FINDER can make a statement about the common law as it relates to the instant case.

None of the systems discussed in this paper (and, to the author's uncertain knowledge, no previously developed system) has incorporated *both* of these methods of legal knowledge representation: a case-based system on top of a rule-based system. It is proposed that such a system would make an ideal focus for LES research. The case-based method used in the FINDER system would be employed so as to provide answers to queries posed when the rule-based method encounters an open-textured problem.

[28] In order for there to be an opposite conclusion, the problem must be one which has only two possible answers. In practice, this should usually be the case as most of these questions can be phrased so that they can be answered "yes" or "no". The FINDER system was concerned with determining the ownership of a chattel which is found by one party, and claimed by another. Even if the problem were to involve three parties (a possibility not addressed by Tyree *et al.*) the question "Who owns the chattel?" becomes "Is the finder the owner of the chattel?".

This approach has a number of benefits:

- It has the advantage of approximating the approach which a lawyer would take when given a legal problem. The rules (derived from a statute) are applied until the meaning of some (open-textured) concept is required. Faced with this problem, a lawyer would turn to the common law in order to further clarify the meaning of the statute. So, too, would this proposed system: the lawyer's two-stage approach would be clearly modeled.

  This similarity to a lawyer's reasoning would be largely irrelevant except that it is the reason that the LES would produce output which could be easily used by a lawyer.

- It goes some of the way towards responding to the complaints of those who believe that an expert system can never adequately simulate legal reasoning.[29] By taking the search for meaning to the common law, this approach avoids some of the problems inherent in a purely rule-based system.

- It is (it would seem) a novel approach to the problem of knowledge representation in LESs. However, the two disparate methods upon which it relies have been separately, experimentally proven.

- It is not subject to the restrictions on the problem domain which bind previous systems to areas of law which are predominantly statute-based, or case-based, but not both.

Choosing a manageable legal problem domain is crucial to the success of any LES project. For an experimental LES which incorporates both a rule-based and a case-based system, the chosen area of law should be covered by only one statute and be the subject of a significant, but not enormous, body of case law.

6 Conclusion

This paper has discussed previous developments in LES design and has shown how a rule-based approach is inappropriate if an LES is to be of use to a lawyer. A better approach (combining rule-based methods with case-based methods) has been outlined and it is suggested that LESs which incorporate this approach will prove to be fruitful objects of research.

Acknowledgements

The author would like to thank Robin Stanton, Malcolm Newey, Peter Drahos, Bob Moles and Neil Tennant for their comments and suggestions.

[29] See Moles' comments in §2.1 and §3.

References

D'AMATO, Anthony (1977). Can/should computers replace judges? *Georgia Law Review*, vol. 11, no. 5 (September), pp. 1277–1301. ISSN 0016-8300.

HART, H. L. A. (1961). *The Concept of Law.* Oxford University Press, London.

MCCARTY, L. Thorne (1980a). Some requirements for a computer-based legal consultant. Technical Report LRP-TR-8, Laboratory for Computer Science Research, Rutgers University, New Brunswick, New Jersey (1 July). Appears, with minor abbreviations, as McCarty (1980b).

MCCARTY, L. Thorne (1980b). Some requirements for a computer-based legal consultant. In *Proceedings of the First Annual National Conference on Artificial Intelligence (AAAI-80)*, Stanford University (18–21 August), pp. 298–300.

MEHL, Lucien (1959). Automation in the legal world: From the machine processing of legal information to the "law machine". In *Mechanisation of Thought Processes*, vol. II, pp. 755–787. Her Majesty's Stationery Office, London. Proceedings of a Symposium held at the National Physical Laboratory, 24–27 November 1958.

MOLES, Robert N. (1987). *Definition and Rule in Legal Theory: A Reassessment of H. L. A. Hart and the Positivist Tradition.* Basil Blackwell, Oxford. ISBN 0-631-15342-X.

SERGOT, M. J., SADRI, F., KOWALSKI, R. A., KRIWACZEK, F., HAMMOND, P. and CORY, H. T. (1986). The British Nationality Act as a logic program. *Communications of the ACM*, vol. 29, no. 5 (May), pp. 370–386. ISSN 0001-0782.

SHANNON, David T. and GOLSHANI, Forouzan (1988). On the automation of legal reasoning. *Jurimetrics Journal*, vol. 28, no. 3 (Spring), pp. 305–315. ISSN 0022-6793.

TYREE, Alan L., GREENLEAF, Graham and MOWBRAY, Andrew (1988). Legal reasoning: the problem of precedent. In John S. Gero and Robin Stanton, editors, *Artificial Intelligence Developments and Applications*, ch. 16, pp. 231–247. Elsevier Science (North-Holland), Amsterdam. Edited selection of papers to the Australian Joint Artificial Intelligence Conference, Sydney, Australia, 2–4 November 1987. ISBN 0-444-70465-5.

WEIZENBAUM, Joseph (1976). *Computer Power and Human Reason: From Judgment to Calculation.* W. H. Freeman and Company, San Francisco. ISBN 0-7167-0464-1.