Artificial Intelligence and Law **7:** 153–183, 1999.

A hybrid rule – neural approach for the automation of legal reasoning in the discretionary domain of family law in Australia

ANDREW STRANIERI
University of Ballarat,
a.stranieri@ballarat.edu.au

JOHN ZELEZNIKOW, MARK GAWLER and BRYN LEWIS
La Trobe University, Victoria, Australia
{johnz,gawler,lewisba}@latcs1.cs.latrobe.edu.au

Abstract. Few automated legal reasoning systems have been developed in domains of law in which a judicial decision maker has extensive discretion in the exercise of his or her powers. Discretionary domains challenge existing artificial intelligence paradigms because models of judicial reasoning are difficult, if not impossible to specify. We argue that judicial discretion adds to the characterisation of law as open textured in a way which has not been addressed by artificial intelligence and law researchers in depth. We demonstrate that systems for reasoning with this form of open texture can be built by integrating rule sets with neural networks trained with data collected from standard past cases. The obstacles to this approach include difficulties in generating explanations once conclusions have been inferred, difficulties associated with the collection of sufficient data from past cases and difficulties associated with integrating two vastly different paradigms. A knowledge representation scheme based on the structure of arguments proposed by Toulmin has been used to overcome these obstacles. The system, known as Split Up, predicts judicial decisions in property proceedings within family law in Australia. Predictions from the system have been compared to those from a group of lawyers with favourable results.

1. Introduction

The majority of applications of artificial intelligence to legal reasoning have focused on domains of law that are not typically regarded as discretionary. Although every legal domain entails some judicial discretion, there are a number of domains in which judicial decision makers have considerable freedom to interpret statutes or precedent cases in their own way. The Family Law Act (1975) of Australia, in distributing property, is regarded as a discretionary Act in that it makes explicit a number of factors that must be taken into account by a judge in altering the property interests of parties to a marriage, but the statute is silent on the relative importance of each factor. Different judges may, and do, reach different conclusions, even when they agree on facts, because each judge assigns different relative weights to factors.

The perception that discretionary domains, such as property distribution in Australian family law, are fundamentally different from non-discretionary domains suggests that the artificial intelligence paradigms which are applied to problems in these two types of domains should also be different. In fact, Edwards and Huntley (1992) noted a discretionary element in Scottish Family Law and reported that rule-based reasoning is inadequate to model reasoning in that domain. Our own early attempts in the application of rule-based reasoning to determine the percentage of assets an Australian Family Court judge is likely to award each party to a failed marriage (Stranieri and Zeleznikow 1992), suggested that a rule-based approach has limited application to simulate reasoning in family law.

In contrast to rule-based reasoning, the artificial intelligence paradigm known as connectionism is particularly well suited to modelling discretionary domains. Unlike symbolic reasoning paradigms, including rule-based and non-monotonic reasoning, domain knowledge of legal rules and principles is not modelled directly in connectionism. Nor is a select number of leading cases stored for subsequent retrieval and adaptation, as in case-based reasoning. Rather, a connectionist learning algorithm is exposed to data from a large number of past cases. This enables the assimilation and sub-symbolic storage of neural connections which mimic the judicial discretion required to weight relevant case factors.

Artificial neural networks (ANN) have not been popular in legal domains for a variety of reasons. The two most difficult problems are this paradigm's inability to generate explanations for conclusions reached, and the difficulty in assembling training sets of sufficient size and coverage. We claim that connectionism can be useful in law if a series of smaller, interconnected networks are used instead of one larger network and if explanations are generated independently of the process used to infer a conclusion. We view these two requisites fundamentally as knowledge representation problems. Thus, our goal was to discover a knowledge representation that assists in the decomposition of a task into smaller sub tasks and which also enables an independent generation of explanations. The knowledge representation we present that achieves this goal is based on the structure of arguments proposed by Toulmin (1958).

The Split Up program, presented here, integrates neural networks with rules to form a partitioned rule/neural system that reasons in the domain of property distribution upon dissolution of a marriage. Knowledge is represented as a series of arguments based on the structure proposed by Toulmin. In Split Up we apply data from a large number of commonplace divorce case judgements to a connectionist algorithm. The algorithm learns to weight factors in the same way as judges have done in past cases, so that the outcome of future cases can be predicted. Rules, legal principles or precedent cases are not explicitly represented in this paradigm. In the jurisprudence of positivist schools, rules, principles and standards are used to reach a judicial decision. However, for legal realists, implicitly utilised rules and principles may be summoned after a decision has been reached in order to ensure that a decision is just, moral and legally correct. As Llewellyn (1962, p. 58) says:

> It was assumed that the deductive logic of opinions need by no means be either a description of the process of decision, or an explanation of how the decision had been reached.

This shift in the status of legal rules enables realists to study the legal profession on an empirical basis and in a similar fashion underpins our use of neural networks to glean the relative weights judges have applied to factors in past cases. Attempts to elicit the weightings from experts or judges by structured interview is inappropriate on pragmatic and also on conceptual grounds. It is difficult for experts to specify how factors are combined with sufficient accuracy such that their reasoning process may be reproduced in a computer implementation. Family law experts know that judges weigh the factors *future needs* and *past contributions* against each other and indeed, experts are adept at doing this themselves to predict a courtroom outcome, yet articulating exactly how these two factors interact is very difficult.

Central to our approach is the knowledge representation method based upon, but not identical to, the argument structures proposed by Toulmin (1958). Before describing our modification and application of the Toulmin argument structure, we discuss the differences between domains that we call discretionary, such as family law in Australia, and other legal domains.

2. Family Law in Australia: A Discretionary Domain

Most legal theorists accept that some degree of judicial discretion is an inevitable feature of any judiciary. Dworkin (1977) discerns two basic types of discretion available to a decision maker, though MacCormick (1981) argues that analysis is flawed in that the distinction between senses of discretion is one of degree and not of type. Hart (1994) assigned judicial discretion a minor role in his jurisprudence. On the other hand, critical legal studies (CLS) theoreticians, as exemplified by Kennedy (1986), have assigned judicial discretion a prominent role.

Despite the attention focused on discretion by jurisprudential theorists, the concept has received little attention from the developers of legal expert systems. In contrast, the concept of open texture has been frequently discussed. Prakken (1993) provides a comprehensive survey of the way in which open texture as a concept has been used to describe indeterminacy in law. He describes distinct types of situations which are open textured and thus, difficult to resolve as those that involve classification ambiguities, defeasible rules or vague terms.

We believe that the existence of judicial discretion contributes to the open textured nature of law, yet situations that involve discretion cannot be described as instances of classification difficulties, defeasible rules or the presence of vague terms. Rather, the existence of discretion is best seen as a distinct form of open texture. Consider a hypothetical panel of Family Court judges who agree on all the facts of a divorce. Vague terms can be imagined to be interpreted in much the same way by members of this hypothetical panel. There are no classification anomalies and the same principles have been used by all judges. In this scenario, outcomes may still be different because judges apply different weights to each relevant factor. Thus, an additional situation is apparent; one where the decision maker is free to assign weights to relevant factors, or combine relevant factors in a manner that is of his own choosing. This will certainly contribute to the indeterminacy inherent in law.

Section 79(1) of the *Family Law Act (1975)* empowers the Family Court to make orders altering the property interests of parties to the marriage but does not lay down procedural guidelines for judicial decision makers. In practice, judges of the Family Court follow a five step process in order to arrive at a property order:

1. Ascertain the property of the parties.
2. Value all property of both parties.
3. Determine which assets will be paramount in property considerations. This is referred to as common pool property.
4. Determine a percentage of the property to be awarded to each party.
5. Create an order altering property interest to realise the percentage.

The Split Up system implements Steps 3 and 4 above, the common pool determination and the prediction of a percentage split. According to domain experts, the common pool determination task (Step 3) does not greatly involve the exercise of discretion, in stark contrast to the percentage split task (Step 4). Consequently, Split Up implements the common pool determination by eliciting heuristics as directed graphs from domain experts using a methodology we have called sequenced transition networks. This approach is described in Section 10 of this paper. In the following sections we describe our use of neural networks and rules for the fourth task above, the determination of a percentage split of marital assets. The last section of this paper describes a classification scheme we have used to help determine which of the five family law tasks can be accurately modelled using existing AI techniques from those which can not.

3. Percentage Split Determination

The *Family Law Act (1975)* directs a decision maker to take into account the past contributions of each party to a failed marriage in addition to their resources for coping with life into the future. Rather than offering one definition for *contributions* and one for *needs*, the statute presents a *’shopping list’* of factors to be taken into account in arriving at a property order. For example, the age, state of health and financial resources are explicitly mentioned in the statute as relevant factors, yet their relative levels of importance are unspecified.

Although the statute presents a flat list of relevant factors without specifying how these factors relate to each other, we realised that the factors could be placed in a hierarchy. The development of the hierarchy required specific knowledge supplied by domain experts. A hierarchy of 94 factors presented in Figure 1 was elicited. Figure 1 demonstrates that the factors relevant for a percentage split determination (extreme right of figure) are past contributions of a husband relative to those of the wife, the husband's future needs relative to those of the wife, and the wealth of the marriage. The factors relevant for a determination of past contributions are the relative direct and indirect contributions of both parties, the length of the marriage and the relative contributions of both parties to the homemaking role. No attempt is made in Figure 1 to represent the way in which relevant factors combine to infer factors higher in the hierarchy. The hierarchy of Figure 1 provides a structure that was used to decompose the task of predicting an outcome into thirty-five sub-tasks. Outputs of sub-tasks further down the hierarchy are used as inputs into sub-tasks higher in the hierarchy. Solid arcs in Figure 1 represent inferences performed with the use of rule sets whereas dashed arcs depict inferences performed using neural networks.

Figure 2 illustrates the framework for inferring a percentage split outcome with the use of a neural network. This figure expands the factors on the right of Figure 1. The inputs to the neural network (depicted on the left edge of Figure 2) are values on each of the three relevant factors, contributions, future needs and wealth. The neural network's output (on the right) is the value of the percentage split predicted. The inferencing of twenty sub-tasks was performed each with its own neural network, whilst for the remaining fifteen sub-tasks, small rule sets were used.

As mentioned earlier, the principal obstacle to the use of neural networks in the legal domain is that explanations for inferences cannot be directly generated from the inferencing process. We have overcome this problem by embedding the neural network within a knowledge representation framework based on the structure of arguments proposed by Toulmin (1958). However, our argument structure differs from that originally proposed by Toulmin in a number of fundamental ways, as described below.

4. Argumentation

Over three thousand years ago, Aristotle presented two types of proofs.[1] Dialectic proofs concern opinions that are adhered to with variable intensity, the objective being to convince or persuade an audience to accept the claims advocated. The second type of proofs are known as analytic proofs. Analytic proofs differ from dialectic proofs in that conclusions are reached by the application of sound inference rules to axioms.

Perelmen and Obtrechts-Tytecta (1969) reflect that modern logic is almost exclusively concerned with analytic proofs. Dialectical proofs and rhetoric have been relegated to a subordinate, if not insignificant position. According to those authors, this has been due to the notion advanced by Descartes (1650) that science ought

[1] Aristotle in Topics.

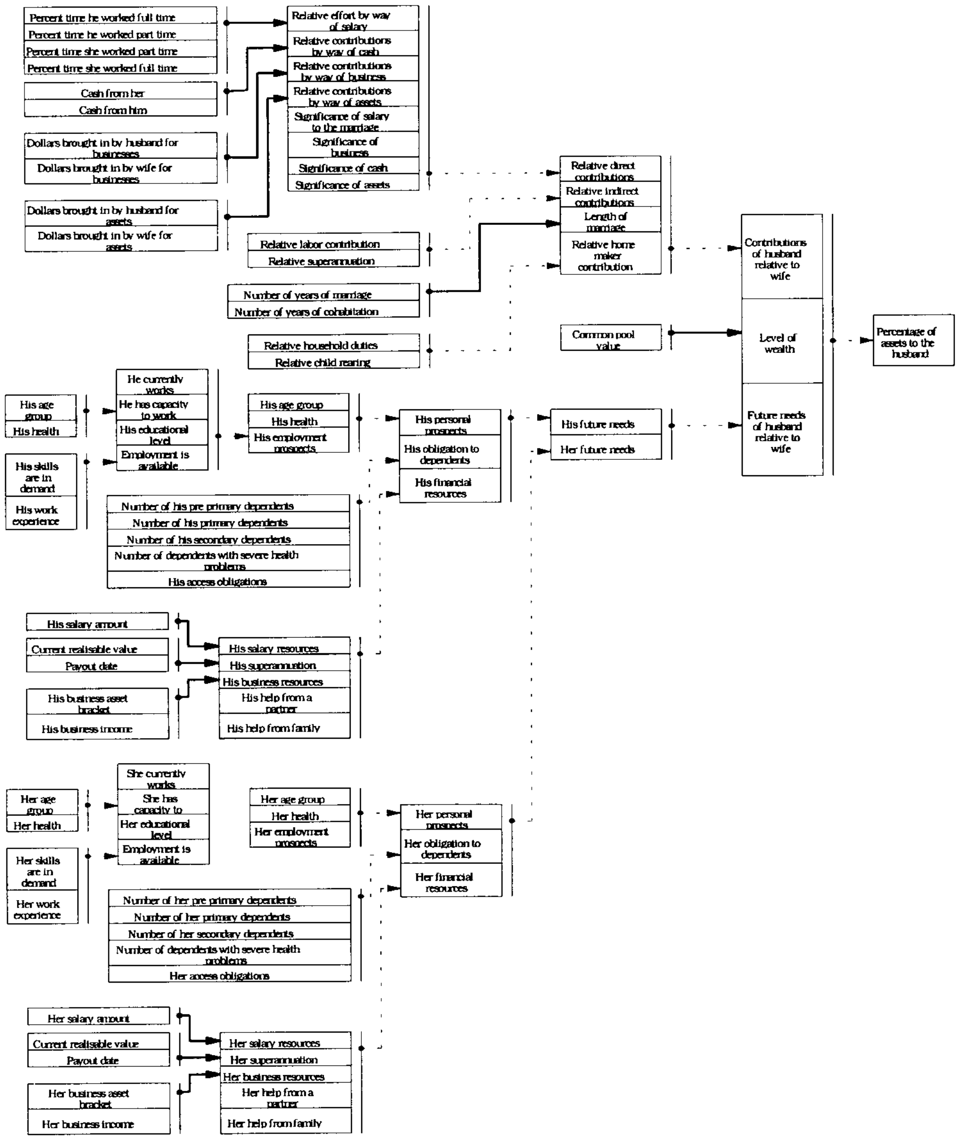

Figure 1. Hierarchy of relevant factors for percentage split determination.

to ignore anything which is based on opinion and cannot be proven. Perelmen and Obtrechts-Tytecta (1969) resurrect the Aristotelian dialectics to the same status as that of analytic logic. Their treatise is entitled ‘The New Rhetoric’ and was originally published in French in 1958.

In the same year, the philosopher Stephen Toulmin published a treatise in English that also sought to resurrect dialectics. For Toulmin, dialectics portrays human reasoning processes far more accurately than analytic reasoning. His treatise on

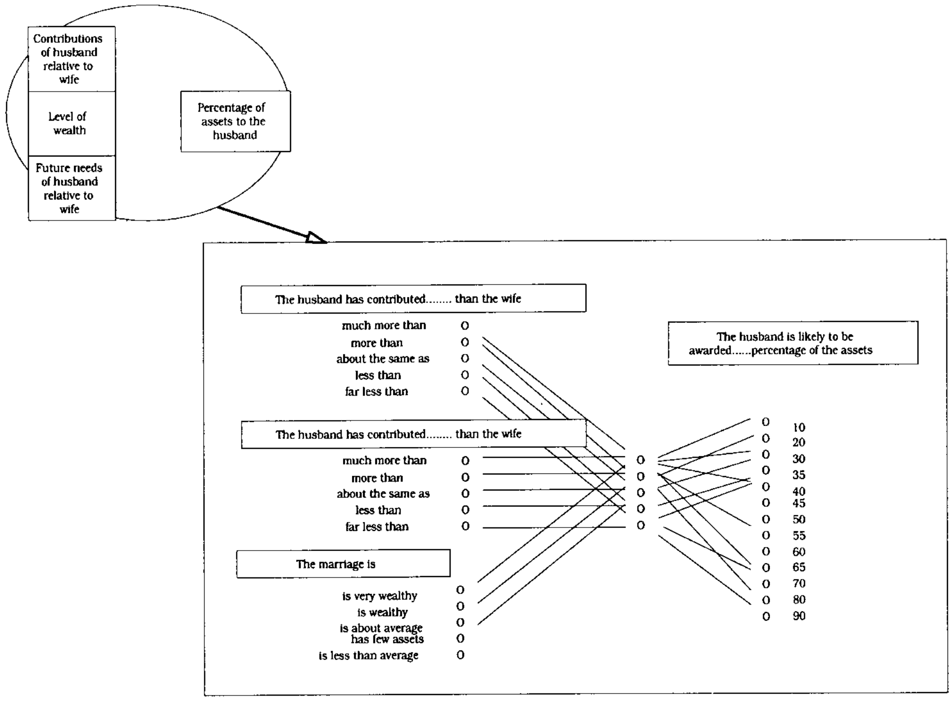

Figure 2. Inferring a percentage split outcome with a neural network.

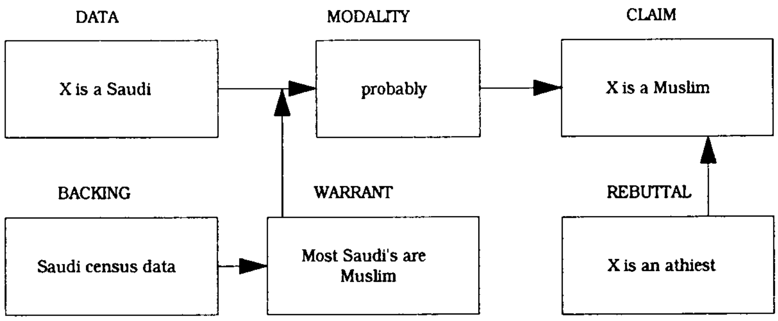

Figure 3. Toulmin argument structure, complete with a sample argument.

argumentation was therefore called a theory of practical reasoning. Toulmin advanced a structure for arguments that was constant regardless of the content of the argument. His treatise focuses on demonstrating that Toulmin argument structures more completely capture the semantics of reasoning than analytic reasoning for the vast majority of arguments. Figure 3 illustrates the structure of argument proposed by Toulmin (1958) with an example that Toulmin uses.

Toulmin (1958) examined arguments from a variety of domains and concluded that all arguments, regardless of the domain, have a structure which consists of six basic invariants: *claim, data, modality, rebuttal, warrant and backing*. Every argument makes an assertion based on some data. The assertion of an argument stands as the claim of the argument. Knowing the data and the claim does not necessarily convince us that the claim follows from the data. A mechanism is required to act as a justification for the claim. This justification is known as the warrant. The backing supports the warrant and in a legal argument is typically a reference to a statute or a precedent case. The rebuttal component specifies an exception or condition that obviates the claim.

Argumentation is a recent phenomena in artificial intelligence that has been used by researchers in two different ways: to structure knowledge and to represent dialectical reasoning. Authors that focus on the dialectical nature of argumentation include Cohen (1985), Poole (1988), Prakken (1993), Gordon (1993), Fox (1986), Farley and Freeman (1995) and Dung (1995). Authors that use argumentation models to enhance knowledge representation include Dick (1991), Marshall (1989), Clark (1991), Johnson et al. (1993), Bench-Capon et al. (1991), Branting (1994) and Ball (1994). The argumentation approach adopted in Split Up falls within this latter group. Figure 4 illustrates all of the components of the percentage split argument in Split Up.

The argument structure we have used differs from the Toulmin structure in three ways:

- reasons which explain why a data item is relevant for a claim are explicitly represented.
- reasons that explain why the inference method used is appropriate are explicitly represented.
- an inference procedure, algorithm or method used to infer an assertion from datum is explicitly represented.

Figure 3 represents the complete argument structure for the percentage split argument. Note that only the claim and data can be represented in the hierarchy of Figure 1.

The reason that the data item "The husband has contributed more to the marriage" is relevant in the percentage split argument within Split Up is that Section 79(4) of the Family Law Act specifically obliges a decision maker to take past contributions into account. Other factors were considered irrelevant and hence are not included within this argument. For instance, there is nothing to indicate that the hair colour of the judge should be considered when deciding the issue of relative percentages. (van Dijk 1989) notes that the notion of relevance has puzzled logicians throughout history and has recently given rise to a class of modal logics broadly described as 'relevance logics'. One aspect of relevance that van Dijk elucidates is the requirement that propositions within the same assertion are expected to be relevant to each other. He eliminates a notion of shared concepts or shared referents as the basis for an understanding of relevance and contends that relevance is firmly rooted in the pragmatics of language. We adopt this stance and

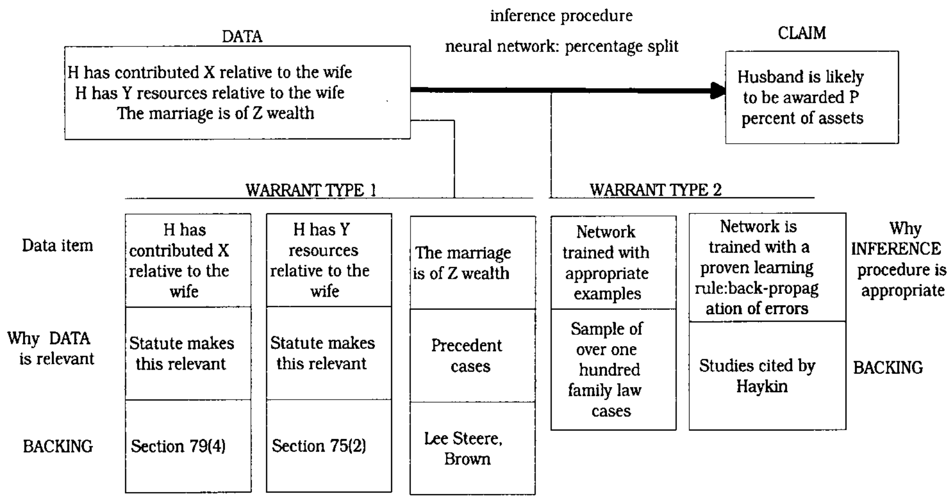

Figure 4. Argument structure used in Split Up.

maintain that a data item is relevant to an argument if a sentence elucidating the reason for the relevance can be uttered.

Explicitly representing the inference method enables the use of a variety of artificial intelligence inferencing procedures. For example, rules are used to infer assertions in Split Up for some arguments whilst neural networks are used for others. Explicitly representing the inference method used in an argument enables us to clearly specify which type of inference has been used for each argument. Argument claims can follow from data by deduction, induction or analogy. The original Toulmin formulation does not permit the specification of the type of inference in use within a particular argument. Knowing the type of inference is important in our efforts to accept or rebut an argument.

A reason that explains why an inference procedure is appropriate is a form of warrant. This contributes to an explanation of why a claim follows from data. As Figure 3 illustrates, the neural network used in the percentage split argument is suitable because it has been trained with data from one hundred and one actual cases. Additionally, it is appropriate that it was trained with a proven learning rule. Conversely, a reason that the application of a rule is appropriate in other arguments is that the inference is an instance of modus ponens, an inference rule that is demonstrably sound.

5. Neural Networks in Law

The use of the connectionist paradigm for modelling legal reasoning is in its infancy. Studies by Bochereau et al. (1991), Walker et al. (1991), Philipps (1989), Bench-Capon (1993), Thagard (1989), Warner (1994) and Hobson and Slee (1994) suggest that connectionism can be usefully applied to resolving open texture in law, though formidable obstacles must be overcome. Concerns that the benefits of neural network use are overstated have been raised by Hunter (1994). These concerns reflect the belief that many applications of neural networks to law have used data that is inadequate because examples are too few in number, or because examples have been invented. Furthermore, the lack of any explanatory facility casts grave doubts on the effectiveness of this paradigm in the legal domain.

We argue that many concerns related to the use of connectionism in law can be allayed. To do so we briefly survey approaches used by past researchers, in the context of the typology of situations characterised as open textured by Prakken (1993). Our aim is to illustrate that neural networks have, in the past, been applied to open-textured situations that are not particularly suited to connectionism. However, they have not been applied to tasks such as those studied here; the mimicking of judicial discretion.

Neural networks have been applied to modelling situations that are characterised as open-textured because of difficulties inherent in performing a classification. Philipps (1989) demonstrates the application of neural networks in dealing with classification difficulties inherent in law with a hypothetical example from Roman Law. He demonstrated that his network generalised well to produce a most reasonable outcome for a case that was not used in the training of the network. He called this an example of the capacity of neural networks to find a solution that represented an equilibrium of past cases, though as Hunter (1994) notes, this notion may be questioned on jurisprudential grounds.

Bench-Capon (1993) also applied neural networks to a problem that involved classification difficulties in the domain of social security entitlements in Britain. The output represented whether or not an applicant was entitled to social security benefits. Open texture manifests in this domain as difficulties inherent in classifying a case. Presented with cases that were not used for training, the network was able to suggest an outcome that reflected the weightings of input variables in prior cases. However, limitations were apparent in that the network was clearly in error in some cases. For example every training set case that output a social security benefit had, as one of the inputs, the fact that the applicant was over 65 years of age. This was because the applicant's age was a limiting condition for the granting of a benefit. However, in some unseen cases, the network granted a benefit to some applicants who were aged under 65 years. In these cases, the 'equilibrium', to use Philipps notation, was achieved in a manner that was quite inappropriate.

Warner (1994) does not explicitly claim that neural networks have the potential to resolve situations in law characterised by classification difficulties. Nevertheless, the actual task he applies them to, in the domain of contract law, appears to be one that attempts to deal with classification difficulties. His network endeavours to classify a case according to whether the contract involved a consideration or not.

Each of the above mentioned authors uses neural networks to resolve classification difficulties in a similar fashion. In contrast, Thagard (1989)'s application of neural networks to legal reasoning attempts to apply connectionism to resolve defeasible rules.

Thagard (1989) proposes a theory of explanatory coherence that aims to model the way in which competing hypotheses are supported to a greater or lesser extent by available evidence. Some nodes in the network he has developed represent propositions that represent each hypothesis. Other nodes represent evidence available. Links which have an associated weight that may be excitatory or inhibitory are defined between evidential nodes and hypothesis nodes. To determine which hypothesis has more support, the network is activated. Nodes feed activation (or inhibition) to other nodes which feedback to each other until an equilibrium is reached. The network is then said to be settled. Thagard trialed his ECHO program on a murder case in which competing hypotheses were *X was innocent* and *X was guilty*. Propositions associated with these hypothesis included *C broke his hand punching X* and *C broke his hand falling on a rock*, respectively.

Thagard's propositions did not include rules from statutes or from legal principles. Even so, there is no reason why this could not have been done. Propositions that reflected statutes or principles would compete for activation with other propositions and those hypotheses that remained most active after the network settled would be deemed to have, in Thagard's terms, more explanatory coherence. In this way, the Thagard approach can be interpreted as one which attempts to resolve those situations in law that are characterised as open-textured because of the presence of defeasible rules. The Thagard approach is certainly intuitively appealing though a great deal of further research is required in order to explore it more fully.

Law is replete with terms which are vague, though few artificial intelligence systems have been developed which reason with vague terms. To our knowledge, connectionism has not been applied to tasks which involve vague terms such as *beyond reasonable doubt*, or *within reasonable limits.* Indeed, it is difficult to imagine how this paradigm can be usefully applied in these situations.

6. Neural Networks in Split Up

We view the use of neural networks in the Split Up project as an exercise in knowledge discovery. Frawley, Piatetsky-Shapiro and Matheus (1991) define knowledge discovery in databases (KDD) as the *'non trivial extraction of implicit, previously unknown and potentially useful information from data'*. Fayyad, Piatetsky-Shapiro and Smyth (1996) highlight that effective knowledge discovery involves a number of steps prior to the application of neural networks. The first phase of any KDD process involves the selection of a sample of data from a store of real world data. In the next phase the data must be pre-processed to remove excessive noise and mistakes. Data is further required to be transformed so that spurious attributes do not clutter the learning algorithm. Neural networks and rule induction are techniques that can be applied in the next phase, known as data mining. Each phase of knowledge discovery in Split Up has required assumptions that we believe are applicable to knowledge discovery in legal fields other than family law.

6.1. PHASE 1: GATHERING RAW DATA

In order to discover how judges weight different factors, we use, as source material, written judgments handed down by judicial decision makers in common place cases. Dick (1991) points out that written judgments of law cases are not transcripts of the arguments presented to a Court during litigation. Rather, the ratio decidendi encapsulated in a written judgment is best seen as an argument that a judge uses to support the decision he has reached. We similarly adopt this view and treat each written judgment as a hierarchy of arguments a Family Court judge has used in deciding a case.

The vast majority of cases that come before the first instance decision maker are never published, are never appealed, and constitute cases that set no precedents. They do not revolve around a new legal interpretation, nor do they involve circumstances that are legally interesting. Zeleznikow et al. (1997) call these commonplace cases and distinguish them from landmark cases. Landmark cases typically set a precedent and are certainly published. Such cases cannot be used for our purposes because our intention is to apply neural networks to learn how judges combine factors in actual day to day practice. As Hunter (1994) and also Aikenhead (1996) note, landmark cases often introduce new law, often revolve around subtleties of interpretation that are far removed from the day to day practice of existing law and are therefore not suited as data sources for neural network training.

In contrast to leading case judgements, written judgments of common place cases are suitable data sources for training neural networks. The argument structure depicted in Figure 1 served as a template for the collection of data from transcripts of actual case judgements. We had access to four hundred family law cases stored within the Melbourne registry of the Family Court of Australia. However, as the focus of Split Up is solely property distribution and many of these cases involved custody issues in addition to property, they could not be used. Expert opinion indicated that property proceedings are certainly influenced by custody matters. One hundred and three cases involved property alone. Three raters extracted data from these cases by reading the text of the judgment and recording values of 94 template variables. Inter-rater agreement tests were performed informally. Any variable that seemed ambiguous or unclear was highlighted so that a consensus could be reached between the raters.

Data for the Split Up project was gathered from cases decided between 1992 and 1994. Each of the cases examined had been decided by one of eight different judges. Judgements from these eight judges were examined in preference to limiting ourselves to those from only one judge in order to encourage the network to mimic a composite of all judges.

6.2. PHASE 2: PRE-PROCESSING RAW DATA

Data from the domain of property division within Australian family law differs from many other domains in that we expect contradictions. For the purpose of our work, we define the term thus: *Two cases are contradictory if their inputs are identical yet their outputs differ*. Contradictions are expected because the weighting of factors can vary between judges and within the same judge over time. Thus, two cases could be recorded with the same input set values but different output values. According to Haykin (1994), contradictory data can severely interfere with the ability of a network to perform adequately on cases that it has not been exposed to.

Contradictory cases are necessarily present in discretionary domains because judges cannot be expected to weight factors in the same way on every case throughout their career, and they cannot be expected to be perfectly consistent with the weightings other judges use. Although contradictory examples are expected in this discretionary domain they should not simply be ignored when training neural networks. A simple example may illustrate this. Consider two cases, A and B, that have identical inputs yet case A resulted in a 70% determination and case B (made perhaps erroneously by a different judge) resulted in 40%. A network trained only with these cases, and presented with identical inputs, will output a value intermediate between the outputs; in this case 55%. The intermediate result of 55% is unacceptable to us. There are a number of ways to deal with extreme contradictions:

- Ignore the extreme contradictions. If sufficient data is collected then the majority of typical outputs will outweigh the effects of a handful of extreme cases. This strategy is acceptable though relies on the existence of quite large data sets for network training. Given the limited sample size, we opted against this strategy.
- Modify one or more contradictory examples to remove the anomaly. This is tantamount to inventing data and was not done.
- Remove extreme contradictions from the training set. This is the strategy we have adopted in this study but we do note that this is not without ramifications.

The strategy of removing extreme examples from the training set can be criticised on both jurisprudential and pragmatic grounds. We argue that although both criticisms are valid their impact is not sufficiently damaging to prohibit the use of this strategy. In domains that are not appreciably discretionary in the way that family law is, it could be said that if two judges arrive at different conclusions after a finding of identical facts then they are using different legal principles or standards. A discussion relating to which of the judges has selected and applied the most appropriate principles opens up intense jurisprudential debates that probably cannot avoid revisiting the discourses centred around the concept of an ideal judge advanced by Dworkin (1977). However, as illustrated earlier in this work, two judges in family law could conceivably agree on the facts of a case and also on the appropriate legal principles yet still reach different conclusions. This is because the principal statute affords the first instance decision maker flexibility in the weighting and combination of factors.

If two judges, in complete agreement on all relevant facts and legal principles arrive at vastly different outcomes in a discretionary domain then we take the view that at least one outcome is in error and consequently must be removed. The error is not an appealable error in law because we assume that both judges have taken into account all relevant facts, and none that are irrelevant. Rather, one (or both) outcomes is an error only in our subjective opinion.

Although the removal of extreme cases from training sets is necessarily a subjective exercise, we can implement a degree of consistency in our method by designing a metric that discerns the extent to which two outcomes are contradictory. The metric we have used in Split Up relies on the representation of all inputs and outputs as binary digits. For example, the percentage split neural network output is not one output that can take any value between 0 and 100 but is, instead, 13 separate outputs each of which can take the value 0 or 1. The same network has 15 binary inputs which represent one of five possible values on three variables.

Two binary outcomes can be compared by noting the position of the set bit in each outcome. Thus, an outcome of 1 0 0 0 0 differs from one represented by 0 0 0 0 1 by four place units. The second set bit is four places away from the set bit in the former outcome. We call this a four place contradiction.

In all networks in Split Up we have removed all examples that have identical inputs but differ from each other by outputs three or more places apart. This criteria is necessarily subjective. Allowing extreme contradictions to remain in the training set is unwise, yet determining which contradictions ought to be labelled extreme is not straight forward.

Table 1 illustrates the number of contradictory examples removed from each training set using this heuristic. Although the claim of twenty arguments are inferred from data values with the use of neural networks only thirteen different networks were used. Seven arguments draw inferences about the husband and have similar counterparts in another seven arguments that draw inferences about the wife. For example, an argument that infers the husband’s obligation to dependents has a counterpart in a quite independent argument that infers the wife’s obligation to dependents. Domain experts were generally of the opinion that the same inferencing procedure in arguments about men could be used as arguments that infer conclusions about women.

6.3. PHASE 3: TRANSFORMATION DATA

The third phase of the knowledge discovery process involves transforming the processed data set to a form likely to be most fruitful. This phase in Split Up involves the decomposition of the task into thirty five sub tasks according to the hierarchy of

Table I. Number of examples for each network with and without extreme contradictions removed.

Net No.	Network name	Description	Number of examples collected	Number of contradictions removed	Training set size
1	Percentage Split	Percentage of assets likely to be awarded to the husband	103	2 (2.06%)	101
2	Relative contributions	Overall contributions to the marriage by the husband relative to those of the wife	103	3 (2.91%)	100
3	Relative needs	Future needs of the husband relative to those of the wife	103	2 (2.06%)	101
4	Relative direct contributions	Contribution made in a direct way to the marriage of the husband relative to those of the wife	103	2 (2.06%)	101
5	Relative indirect contributions	Contribution made in an indirect way to the marriage of the husband relative to those of the wife	103	15 (14.56%)	88
6	Relative home maker contributions	Contribution made as a homemaker by the husband relative to those of the wife	103	0 (0%)	103
7	Individual needs	Extent to which an individual has a need for resources in the future	206	5 (4.85%)	201
8	Individual personal prospects	Future prospects based on personal skills, abilities, age and health	206	29 (14.07%)	177
9	Individual employment prospects	Employment prospects for an individual in the future	206	9 (4.36%)	197
10	Individual capacity to work	Capacity an individual has to engage in future employment	206	41 (19.9%)	165
11	Availability of employment opportunities for an individual	Likelihood of employment opportunities in the future for an individual	206	4 (1.94%)	202
12	Individual financial resources	Extent of financial resources available to an individual	206	60	146
13	Individual business resources	Extent to which an individual has recourse to resources from investments or businesses	35	5	30

arguments depicted in Figure 1. Each sub task could thus be treated as a separate (and smaller) data mining exercise. This decomposition also enabled each set of examples to be free of null values.

6.4. PHASE 4: DATA MINING USING NEURAL NETWORKS

Data mining was performed in Split Up with the use of neural networks. There are many types of network that could have been used, though we restrict ourselves to feed forward networks trained with backpropagation of errors. We claim no new contribution to the general field of neural network research and hence used the type of network that has been the most popular. We restricted ourselves to networks with one hidden layer because these networks are simpler than networks with multiple hidden layers and, as Cybenko (1989) demonstrated, any continuous function can be approximated with single hidden layer networks. The optimal number of units in the hidden layer of any network is difficult to determine. Future work aims to draw on programs such as that developed by Lengers (1995) which applies genetic algorithms to determine the optimal topology. However, for the current work, we aimed to keep the number of hidden units down to as small a value as possible.

The aim of any classifying system is to be able to classify all cases in the domain in question. In the domain of family law property proceedings this includes every case that has been heard in the past and every case that will be heard in the future. Access to the entire population of cases is clearly impossible. However, if a random sample of cases is selected from the entire population and we apply appropriate statistical techniques, then it is possible to estimate the error our classifier systems will make over the entire population. Thus, particular attention must be focused on the production of estimates of the extent to which our networks generalise to the true population of cases. In the next section, we discuss our efforts to train neural networks so that they can be accurate predictors for any case.

Feed forward neural networks trained with backpropagation learning are said to generalise well if the output of the network is correct (or nearly correct) for examples not seen during training. According to Haykin (1994), generalisation is influenced by the size and coverage of the training set, the architecture of the network and the complexity of the problem. Two extremes are to be avoided if adequate generalisation is to be achieved; under-training and overtraining.

As the term suggests, under-training occurs if the network is exposed to too few examples. Learning is difficult in this situation because the training patterns available are not sufficiently representative of the true population of cases or because the network has not been exposed to the training set for a sufficient number of repetitions (epochs). The opposite extreme is known as over-generalisation or overtraining. If a sufficiently large network has been exposed to too many examples, too many times, it can learn each input-output pair so well that it, in effect, memorises those cases it has been exposed to. The network is said to be overtrained and does not generalise well.

Our objective in training Split Up neural networks was to avoid these two training extremes. This was achieved, in part by terminating training at an appropriate time. The criteria used to decide when to cease training is related to the metric used to measure a network's performance. A performance measure typically used in network training in non-legal domains is the proportion of examples correctly classified. As Weiss and Kulikowski (1992) point out, this measure of network performance may not be adequate for all domains. They suggests that a metric that includes the direction of the error is useful in some domains. For instance, classifiers in medical domains will ideally rarely fail to detect a disease though may frequently err in raising a false alarm. A cost/risk classification of errors is not warranted in family law because the direction of the error is rarely as critical as it is in medical domains. A network that predicts the husband is to receive 60% of the property errs if the judge in the case actually awarded 55%. However, another network errs in a similar and in a no more or less damaging manner if it predicts the husband is to receive 50% of the assets. Thus, the direction of the error is not critical for our purposes. Nevertheless, the magnitude of the error warrants special attention.

Simply counting the number of correctly classified examples leads to a measure of network performance which may be too fine-grained for legal applications and increases the risk of over training. We argue that a better measure of a network's performance includes an indication of the magnitude of the error. For example, a variation of 5% either way from a judge's decision of the percentage of assets to be awarded to the wife is, in our view, a minor error. On the other hand, a network which outputs a percentage split which deviates from that obtained by a judge by 20% is assumed to have erred. Although the cut off point for declaring that an error has occurred is necessarily subjective, it was important that a metric be discerned which could be applied consistently to all networks in Split Up.

To our knowledge, no application of neural networks in law, using real or hypothetical data, have employed techniques to ensure that a trained network reflects patterns in the actual population of cases and is not merely a reflection of the sample data gathered. Weiss and Kulikowski (1992) provide a comprehensive overview of statistical techniques that help to ensure a trained network represents characteristics of the entire population and is not an aberration linked solely to the particular set of cases selected for training and testing. They note that few classifiers remain generally accepted unless some effort has been made to evaluate the performance of the classifier on an entire population and not just on a sample of data. In the next section, we shall summarise techniques used to extrapolate classifier performance on sample data to an entire population.

The practice known as hold-out involves partitioning the sample data into a training set and a test set. The classifier is trained with the training set while the test set is held out. Once trained, the classifier is trialed on the test set. The error rate on the test set provides a better estimate of the true error rate than the apparent error figure. Hold-out is not ideal in a domain such as law which is characterised by a lack of readily available data because examples held out must be prohibited from contributing to the training. Techniques known as re-sampling provide methods for estimating the true error rate without denying the learning system access to valuable data.

6.5. RE-SAMPLING TECHNIQUES

Weiss and Kulikowski (1992) describe cross validation as a technique that involves randomly generating a number of training/test set partitions. The test set in each training/test pair is held out while the classifier is trained on the training set. The estimate of the error rate on a true population is determined by taking an average of the error rate on each test set. An issue that arises with the use of cross validation concerns the optimal size of each training and test set.

K-fold cross validation involves partitioning the sample into *k* training/test set pairs of approximately equal size. *k* classifiers are trained and the estimate of the true error is obtained by taking the average of apparent error rate on the *k* test sets. Cross validation is the most popular re-sampling technique in computer learning systems though it has disadvantages particularly with small samples. We chose to use *5 fold cross validation* because of its popularity although it is worth noting that the technique known as bootstrapping warrants special attention in future research in law because of its potential to handle small data sets. Table 2 represents the topology and performance of neural networks in Split Up. Results depicted in that table are the average number of errors of each magnitude over the five cross validation sets.

7. Implementing Split Up

Split Up has been implemented using KnowledgePro[2] as an argument-based reasoning shell. Family law knowledge has been entered into the shell so that the argument-based framework can be evaluated, though studies are under way to demonstrate that the shell can also be useful applied within non-legal domains. The basic unit of knowledge in Split Up is the sentence. All data, claim, warrant and backing items are sentences. All sentences are stored in a sentence base and are retrieved to produce user prompts, claims and explanations. Arguments are frames with slots that reference sentences in a sentence base.

An argument sequence that corresponds to the tree of Figure 1 is not stored explicitly. Instead, the sequence of arguments that is required for a percentage split assertion is created dynamically at the outset of a consultation. We call this a chain of reasoning and note that its dynamic creation facilitates future maintenance.

[2] KnowledgePro is an object oriented high level language with a built in inference engine and hypermedia development tools released by Knowledge Garden Inc. and runs on a PC-Windows environment.

Table II. Performance of Split Up networks

Net No.	Network name	Topology input-hidden-output	Average proportion of errors of magnitude >3	Average proportion of errors of magnitude >2	Average proportion of errors of magnitude >1	Average proportion of errors of magnitude >0.5	Number of epochs
1	Percentage Split	15-12-13	0.03	0.12	0.16	0.31	900
2	Relative contributions	20-8-5	0.01	0.02	0.06	0.27	1130
3	Relative needs	8-3-5	0.00	0.01	0.02	0.07	230
4	Relative direct contributions	32-15-5	0.00	0.01	0.09	0.19	35
5	Relative indirect contributions	10-4-50.02	0.00	0.02	0.13	180	
6	Relative home maker contributions	10-5-5	0.09	0.00	0.00	0.00	270
7	Individual needs	14-3-4	0.02	0.07	0.11	0.30	830
8	Individual personal prospects	17-9-5	0.02	0.01	0.05	0.26	480
9	Individual employment prospects	14-8-5	0.02	0.02	0.08	0.101	170
10	Individual capacity to work	12-5-3	0.01	0.00	0.06	0.22	180
11	Availability of employment opportunities	9-5-4	0.00	0.12	0.16	0.4	490
12	Individual financial resources	26-12-5	0.03	0.02	0.11	0.31	180
13	Individual business resources	10-5-5	0.02	0.03	0.29	0.40	160

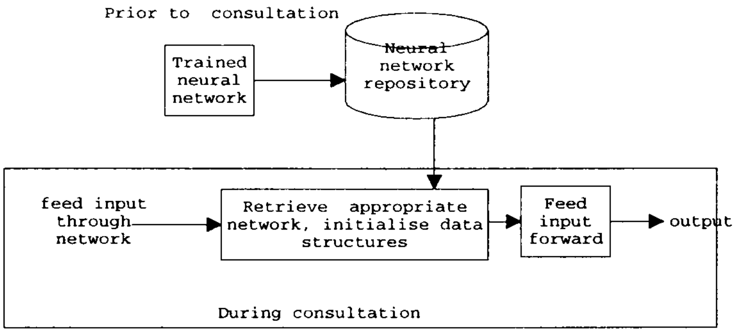

Figure 5. Schema for typical neural systems.

Once the chain of reasoning is constructed inferencing proceeds by repeating the following steps until the final argument is executed:

- retrieve the next argument from the chain of reasoning;
- if a value for the argument's data value is not known, then prompt the user; and
- retrieve the inference method for that argument from the argument base and apply the data items to the inference method;

Software used for neural network training was neuDL (Neural Network Description Language), a description language for the design, training and operation of neural networks developed by Samuel Joe Rogers at the University of Alabama. Using this language rather than specialised network software enabled us to implement our own performance metrics described above. All neural networks were trained on mainframe computers running Unix for speed and efficiency.

Invoking a number of neural networks for a single consultation presents an efficiency problem. Figure 5 represents the system design typically used when a number of different neural networks are called upon to perform an inference. This figure illustrates that once trained, each network's topology, weights, biases and activation function are stored in a repository. A run time invocation of a network requires, in essence, that the network be rebuilt with information retrieved from the repository before the input data can be fed through to produce an output. This can seriously degrade performance in a system such as Split Up that consecutively invokes over twenty networks. Rather than include the neural network functions in Split Up, we have captured the results of inferencing in a data structure.

Each possible input is presented to the trained network prior to a consultation in a pre-specified order. The position of a given input in this ordering is determinable solely from the input's value. The outputs are stored in the order they emerge from the network, so that the outputs are also ordered. To find the output which corresponds to given input, the position of the input is determined and the output at that position in the output list is retrieved.

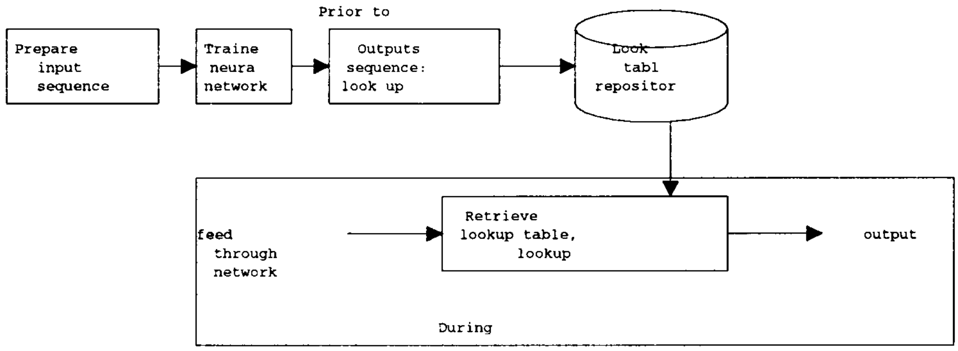

Figure 6. Schematic diagram of lookup table repository method.

This approach is illustrated in Figure 6. There is no need to store information about each network other than the sequence of outputs. Rebuilding a network is unnecessary, nor is there a need to actually feed values through a network during consultation. Instead the output value that corresponds to the input is retrieved from the ordered list of outputs. These factors combine to greatly enhance the efficiency of the system.

8. Explanations in Split Up

Split Up explains its reasoning for inferring an argument's assertion by presenting the data, warrant and backing components of the argument to the user on request. For example, if the user invokes an explanation for the assertion *"Overall, the husband is likely to be awarded 40% of the assets"* he/she is presented with the data items from the argument structure "*The husband has contributed to the same extent as the wife, The husband has greater resources for the future as the wife, The marriage is of average wealth*". If any one of these data items are questioned by the user, the argument that produced the data item as an assertion is retrieved and an explanation generated from it. If, on the other hand, the user is satisfied with the data items but wants further explanation, the reasons for the relevance of each data item and the reason for the appropriateness of the inference method are retrieved from the argument structure: "*Contributions must be taken into account according to the statute: 79(4). Resources for the future must be taken into account according to the statute: 79(4)e 75(2). Inference has been produced using a neural network trained with appropriate examples: over one hundred real Family Court cases. The neural network was trained using backpropagation of errors: a proven learning algorithm".*

Eberhart (1995) claims that the purpose of an explanation facility within an expert system is to encourage the user to trust the system as opposed to the purpose of rule based system explanation facilities which was to aid knowledge engineers to debug large rule sets. The early 'trace' type of explanation facility reflected the inferencing process perfectly though typically did not engender a user's trust. In a similar vein, Bench-Capon et al. (1991) noted that explanations were more than proof procedures and reported favourable user responses when they used Toulmin argument structures to provide explanations for their logic programs. Wick and Thompson (1992) also note that explanation involves more than the reproduction of inferencing steps. They have developed an explication facility that is invoked after the inferencing has concluded. It takes, as input, the inferencing steps used to reach a conclusion, in addition to domain knowledge at a different level of specificity to that used to infer conclusions.

Generating explanations for neural network inferences is difficult because the inferencing steps are not explicit. Nevertheless, two broad approaches have emerged which aim to generate explanations in this paradigm. One approach involves selecting a sample of examples that most closely matches the input. These examples are presented to the user as similar cases. This approach certainly has its uses but can be limited in law. A sample of similar examples does not make explicit a statute or precedent case that underlies many inferences. A different approach in neural network explanation, exemplified by Diederich (1992), involves representing the internal sub-symbolic processes within a neural net in a symbolic manner so that inference steps can be elucidated. While useful, these approaches are limited in law because an explanation that will engender trust must provide information over and above that involved in inferencing steps.

Explanations in Split Up are pragmatically grounded because they are supplied by domain experts. There is no suggestion that the reason for the relevance of a data item specified by domain experts is the only one possible, nor is it necessarily the ideal reason. It is however a reason that makes pragmatic sense to the expert. As such, it is more likely to engender the user's trust than if a reason which replicated reasoning steps was used.

9. Split Up Evaluation

Eight specialist family law solicitors were asked to analyse three cases.[3] The three cases were devised to test diverse marriage scenarios. Given the difficulty in assembling large numbers of specialist lawyers we cannot attempt tests of significance on these results. Table 3 illustrates the percentage of the assets awarded to the husband by the Split Up system and by each of the lawyers.

Cases B and C indicate compatibility between Split Up predictions and those of the eight lawyers. Furthermore, reasons for their prediction given by Split Up were similar to reasons given by the lawyers. Case A was more controversial. This case involved a marriage where domestic duties were performed by paid staff and not by either party to the marriage. Split Up and four of the lawyers interpreted this situation as one where both parties had contributed to the home in equal measure.

[3] Ten lawyers actually participated in the trials but results from two were not included because of signficant prior awareness of the Split Up system.

Table III. Percentage of assets awarded to husband by Split Up and family lawyers

	Case A	Case B	Case C
Split Up	55%	50%	40%
Lawyer 1	55-60%	50%	35%
Lawyer 2	55%	50%	35–40%
Lawyer 3	50–55%	50%	40%
Lawyer 4	45%	50%	50%
Lawyer 5	45–50%	50%	40%
Lawyer 6	40%	50%	35%
Lawyer 7	45–50%	50%	35%
Lawyer 8	50%	50%	40%

The remaining lawyers regarded this situation as improbable and, despite evidence to the contrary, assigned the majority of the home-maker role to the spouse who had not engaged in paid employment. This brings to light an important issue in this legal domain: What constitutes being a home-maker, and how important is this role within a marriage? As the above example in Table 3 clearly shows, the domestic beliefs of the lawyers involved weighed heavily on their interpretations of the given facts. Evaluating the Split Up system is complicated by such variance within domain experts' opinions.

10. Sequenced Transition Networks

The sequenced transition network methodology (STN) enables the automated translation of a directed graph into sets where each set represents a path within the graph (Stranieri et al. 1994). Four set operators defined in the STN approach are applied to the sets in order to implement forward chaining, backward chaining and the generation of explanations. This approach is conceptually equivalent to rule-based reasoning although the role of a knowledge engineer is kept to a minimum. Using this methodology, knowledge acquisition and maintenance benefits result because rules are not required at all. There is no requirement to convert graphs to rules because directed graphs drawn by the expert are automatically converted into sets. Figure 7 illustrates a directed graph that represents the interaction between a family law expert and a client.

The principal domain expert for Split Up was Renata Alexander, an experienced family law practitioner with a government funded legal agency in the state of Victoria. The graph of Figure 7 is one of 51 she drew to capture knowledge relevant for determining whether an asset will be considered by the Court or not. An *STN* program labels each node in the graph. The initial node is labelled 0. The node

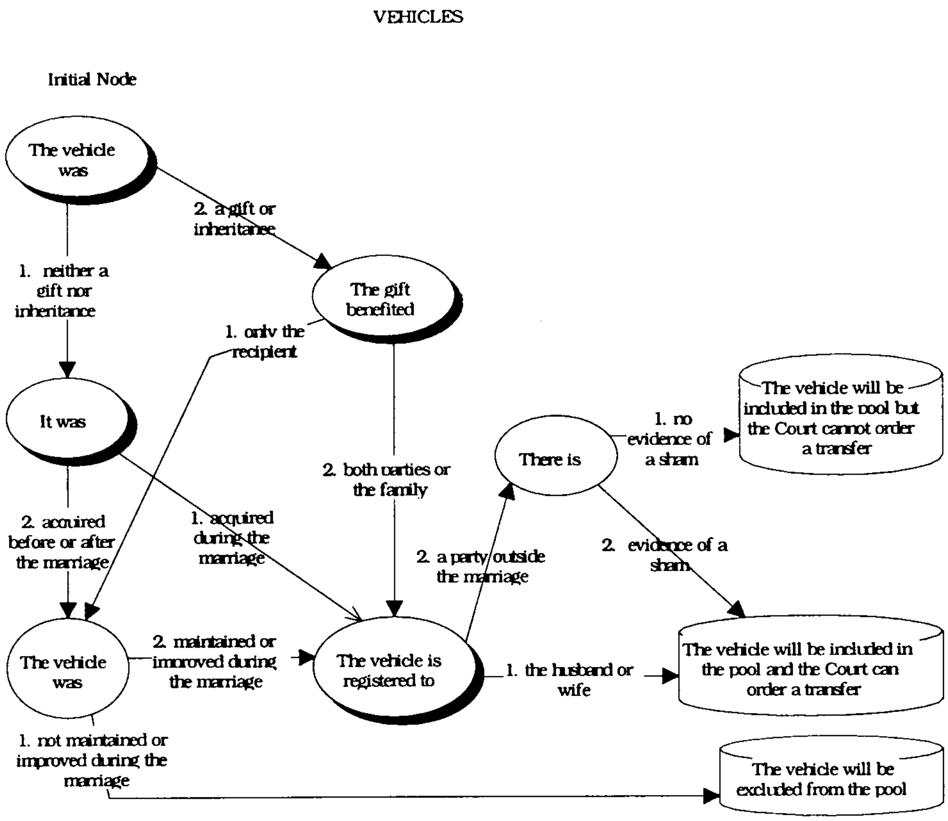

Figure 7. Directed graph for common pool determination.

reached by traversing arc 2 from the initial node is labelled node 02. If a node in the graph is reached by more than one path, then the node receives two labels. The program stores each node as an object which is identified by the node label. Text associated with the node and with arcs emanating from the node on the graph are also stored within the object for subsequent use as prompts and explanations. Thus, a directed graph is not translated into rules for inference engine utilisation as is the case in traditional rule-based expert system design. No prompts, questions or text other than what is drawn on the graph are required.

Forward chaining inferencing commences with the presentation to the user of the text associated with the initial node as a user prompt. The user is then provided with a number of response options. The arc number of the user's response is appended to the node number to retrieve the next node. This proceeds until a conclusion is reached. Three set operators are invoked to generate explanations:

- Reachable conclusion set for a node. This is the set of all conclusions that can be reached from the node.

- Unreachable conclusion set for a node. This is the set of all conclusions that cannot be reached from the node.
- Difference segment. This is a set that compares two paths and represents the first point of difference between the two sets.

The sequenced transition network methodology is an approach equivalent to a rule based approach but has some acquisition and maintenance benefits. However, a pressing question in the use of this technique, neural networks or any other inferencing method concerns the selection of the problem to be implemented. The next section introduces a classification method we have used to help select those aspects of Split Up's target domain which could best be implemented with rule sets and those best left to neural networks.

11. Evaluating Tasks for AI Implementation

We considered the common pool determination task well suited to a rule-based reasoning approach, but the task of predicting the percentage of those assets awarded to either party was perceived to be too demanding for rule-based reasoning. The problem of deciding whether a legal task can be modelled by any existing paradigm, and if so, which one, is a problem currently tackled in an ad hoc manner by developers of legal reasoning systems. This section describes a simple classification scheme we used to classify sub-tasks in Split Up in an attempt to instil some method within our decision making.

The classification scheme is based on two dimensions; our estimation of the extent to which a task is open textured, and our estimation of the extent to which a task displays a feature that we call boundedness. A graphical representation of four family law tasks, based on two axes named *open textured-well defined* and *bounded-unbounded*, is presented in Figure 8. The dimension *open textured-well defined* refers to our belief as to the extent to which a task is open-textured. Although every possible extension for an open-textured concept cannot be predicted, we believe that it is possible to estimate the extent to which the known extensions represent all possibilities. Practitioners seem to estimate the degree of open texture of a statute in order to offer a prediction. Few practitioners would argue that the concept of *vehicle* in a local government by-law seems less subject to new uses than the concept of *membership of a social group* in a United Nations Convention governing refugee status.

The *bounded-unbounded* dimension refers to the extent to which an expert (or knowledge engineer) believes that all terms relevant for the performance of a task are explicitly known. The aetiology of AIDS is quite unbounded because we suspect that there are factors relevant to this task other than a HIV test result. We have a fair degree of confidence that expert heuristics for the common pool task depicted in Figure 8(a) are relatively complete thus this task is perceived to be

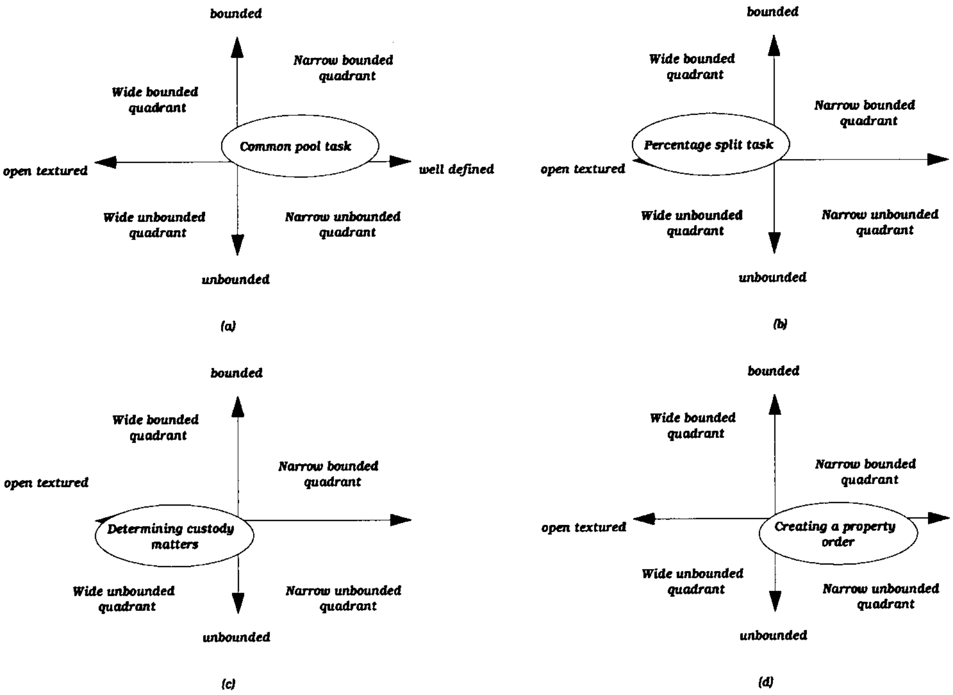

Figure 8. Classification of four family law tasks.

quite bounded. Similarly, we believe we know sufficient relevant factors to predict a property distribution percentage (Figure 8(b)).

The task of predicting custody arrangements depicted in Figure 8(c) is quite unbounded in that we do not believe that all, or even most factors relevant for this determination are known. Practitioners tell us the family values particular judges adhere to is an important factor in predicting custody outcomes yet this is not a factor we can include in an automated system with confidence. Furthermore, the character judgement inherent in determining custody matters necessarily involves factors that cannot obviously be made explicit. Therefore, we are enticed into believing that this task cannot be modelled adequately because sufficient knowledge does not currently exist.

Figure 8(d) depicts the task judges have in creating a property order to realise a percentage split, once decided. Few features relevant for this task are known though judges generally avoid forcing a sale of any asset and they also attempt to minimise the disruption to the everyday life of children. There are no other obviously relevant factors or heuristics. The statute provides no guidance and there have been very few contested cases which specifically relate to the court order created. We thus view this task as quite unbounded.

We believe that the classification of a task along the *bounded/unbounded* continuum is subjective. We highlight three factors that affect an expert's perception of the degree of boundedness in a task.

- Prevalence of discretionary provisions in statutes. These provisions encourage a decision maker to take any factor deemed relevant into account.
- The expert's perception of the broader social and political environment.
- The expert's estimation of the completeness of his/her knowledge.

A classification along the open-texture continuum is necessarily subjective. The same task may be classified by different experts in different ways. This is understandable because the classification reflects the expert's belief about the domain. More specifically, we believe the classification reflects the expert's beliefs about four factors.

- The extent to which the domain contains ambiguous definitions;
- The extent to which terms in the domain are coarse grained;
- The expert's own jurisprudential perspective; and
- The social and political environment.

The common pool task of Figure 8(b) does not seem appreciably open textured to us. There is far less discretion than is the case in the percentage split task represented in Figure 8(a). Heuristics used for the common pool task involve terms which are not obviously susceptible to new uses. In contrast, the discretionary element available to a decision maker in combining relevant terms in the percentage split problem lead us to view this task as quite open textured. Terms relevant for the prediction of a property order include the percentage split figure to be realised, the desire of each party to retain assets and the disruption to children. In our view these concepts, perhaps with the exception of *'disruption to the children'*, are not appreciably open-textured.

Tasks that fall in the narrow bounded quadrant (top right) are well suited to implementation with rule-based reasoning or within a logic programming paradigm. First order predicate logic limits inferences to deduction, is monotonic, and cannot represent uncertainty. In many tasks these features are limitations, but this is not the case with tasks classified as narrow bounded. A representation of uncertainty is not required for these tasks because terms relevant for a solution are known and how they combine is also known. Terms are not appreciably open textured so there is little need to represent uncertainty of rules or clauses. Similarly, non-monotonicity is not necessary because assertions do not normally need to be retracted. The common pool task was therefore implemented as a rule based reasoner, albeit a non-traditional one.

Tasks classified as wide bounded (top left quadrant) are regarded as *bounded* to a considerable extent but do contain many terms which are quite open-textured.

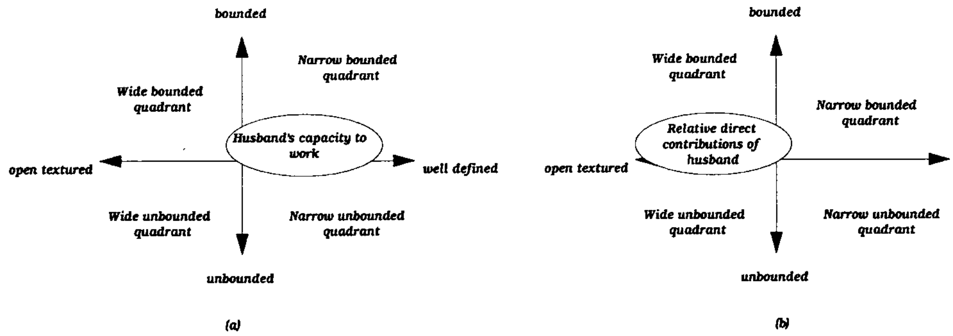

Figure 9. Two arguments within percentage split determination.

Many legal tasks appear to fall within this category. A technique is required to deal with the open texture in these domains. As described earlier, the open texture in percentage split determination involves judicial discretion so we have used neural networks. If the open texture manifests predominantly as defeasible rules, then a non monotonic logic would seem more appropriate.

We believe that unbounded tasks, whether they contain open textured terms or not, cannot be performed by any existing paradigm because relevant terms are not known. The aetiology of AIDS cannot be predicted because sufficient terms relevant for that prediction are not yet known. The task of predicting the actual property order a judge may deliver in order to realise a percentage split of the assets previously decided was considered to be beyond the scope of any existing AI paradigm simply because insufficient factors are known about this decision making process. Predicting custody arrangements is made difficult not only because of a preponderance of open textured terms such as *welfare of the children* but also because we suspect factors vital for the prediction are not known to us at this time.

We have used the classification scheme with sub tasks (arguments) in the percentage split determination task to discern those arguments that are suited to implementation with a rule based approach from those more suited to neural networks. Figure 9 illustrates two tasks within the percentage split determination which are represented as arguments. The husband's capacity to work illustrated in Figure 9(a) is determined by two factors; his age and his state of health (see Figure 1). These terms do not seem appreciably open textured to us, principally because their combination is quite straight forward. These are the only factors that seem relevant for determining a capacity to be engaged in employment. We have therefore classified this task as a predominantly narrow bounded one and have hence implemented it with rules. In contrast, the way the datum items combine to infer a relative direct contribution of the husband depicted in Figure 9(b) involves a degree of discretion and is thus seen as quite open textured. This task has therefore been implemented using neural networks.

12. Conclusion

We have demonstrated our belief that reasoning in discretionary domains is indicative of a type of open-texture often overlooked by AI and Law researchers. This type of open-texture needs to be tackled in a different way to the methods used for classification anomalies, defeasible rules or vague terms. We have illustrated the benefits of an integration of the connectionist paradigm with rule-based reasoning for reasoning in the discretionary domain of family law in Australia. Our approach generates explanations for conclusions which are reached quite independently of inferencing methods used to reach those conclusion. Assumptions underlying this draw jurisprudential support from the movement known as legal realism. The foundation of the integration and the explanation is a knowledge representation schema based on the structure of arguments proposed by Toulmin. We have also illustrated a classification scheme that helped us discern those tasks more suited to a rule based implementation from those suited to a connectionist implementation.

References

Aikenhead, M. 1996. The uses and abuses of neural networks in law, *Santa Clara Computer and High Technology Law Journal* **12**(1), 31–70.

Ball, W.J. 1994. Using Virgil to analyse public policy arguments: a system based on Toulmin's informal logic, *Social Science Computer Review* **12**(1), 26–37.

Bench-Capon T.J.M, Lowes, D., and McEnery, A. M. 1991. *Argument-Based Explanation of Logic Programs*, Volume 4, No 3. Knowledge Based Systems,pp. 177–183.

Bench-Capon, T.J.M. 1993. Neural networks and open-texture. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*. Amsterdam: ACM Press, pp. 292–297.

Bochereau, L., Bourcier, D., and Bourgine, P.: 1991. Extracting legal knowledge by means of a multilayer neural network application to municipal jurisprudence. *Proceedings of the Third International Conference on Artificial Intelligence and Law*. Oxford. ACM Press, p. 288.

Branting, K.L. 1994. A computational model of ratio decidendi, *Artificial Intelligence and Law* **2**, 1–31.

Clark, P. 1991. *A Model of Argumentation and Its Application in a Cooperative Expert System*. PhD thesis. Turing Institute. Glasgow: Department of Computer Science, University of Strathclyde.

Cohen, P. 1985. *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach*. London, Pitman.

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals and Systems* **2**, 303–314

Descartes, R.A. 1650. *Discourse on the Method*, translated by J. Veith, 1912. London: J.M. Dent Pty Ltd. Last printed in 1965.

Dick, J.P. 1991. *A Conceptual, Case-Relation Representation of Text for Intelligent Retrieval.* Ph.D Thesis. Canada: University of Toronto.

Diederich, J. 1992. Explanation and artificial neural networks, *International Journal of Man-Machine Studies*. **37**, 35–355.

Dung, Phan Minh. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* **7**(2), 321–357.

Dworkin, R. 1977. *Taking Rights Seriously*. Cambridge: Harvard University Press.

Dworkin, R.M. 1967. The model of rules. *University of Chicago Law Review* **38**, 14–46.

Eberhart, R.C. 1995. Using evolutionary computation tools in explanation facilities, *International Journal of Expert Systems* **8**(3), 277–285.

Edwards, L. and Huntley, J.A.K. 1992. Creating a civil jurisdiction adviser. *Law, Computers & Artificial Intelligence* **1**(1), 5–40.

Farley, A.M. and Freeman, K. 1995. Burden of proof in legal argumentation. *Proceedings of Fifth International Conference on Artificial Intelligence and Law.* May 21–24. Boston: ACM Press, USA, pp. 156–164.

Fayyad, U, Piatetsky-Shapiro, G., and Smyth, P. 1996. The KDD process for extracting useful knowledge from volumes of data, *Communications of the ACM*. **39**(11), 27–41.

Fox, J. 1986. Knowledge, decision making and uncertainty. In W.A. Gale (ed.), *Artificial Intelligence and Statistics*. Reading, MA: Addison-Wesley.

Frawley, W.J., Piatetsky-Shapiro, G., and Matheus, C.J. 1991. *Knowledge Discovery in Databases: An Overview*. AAAI: MIT Press, pp. 1–27.

Gordon, T.F. 1993. The pleadings game. *Proceedings of Fourth International Conference on Artificial Intelligence and Law.* Oxford, New York: ACM Press.

Hart, H.L.A. 1994. *The Concept of Law*. 2nd ed. Oxford: Clarendon Press.

Hobson, J.B. and Slee, D. 1994. Indexing the Theft Act 1968 for case based reasoning and artificial neural networks, *Proceedings of the Fourth National Conference on Law, Computers and Artificial Intelligence*. Exeter.

Haykin, S. 1994. *Neural Networks: A Comprehensive Foundation.* New York: Macmillan..

Hunter, D. 1994. Looking for law in all the wrong places: Legal theory and legal neural networks. In H. Prakken, A. J. Muntjewerff, A. Soereman, and R. Winkels (eds.), *Legal Knowledge Based Systems: The Relation with Legal Theory*. Lelystad: Koninklijke Vermende, pp. 55–64.

Johnson, P.E., Zualkernan, I.A., and Tukey, D. 1993. Types of expertise: an invariant of problem solving, *International Journal of Man Machine Studies* **39**, 641.

Kennedy, D 1986. Freedom and constraint in adjudication: A critical phenomenology, *Journal of Legal Education* **36**, 518–562.

Lengers, R.J.C. 1995. *Evolving Artificial Neural Networks. A Design Approach*, Masters Thesis. Department of Economics, The Netherlands: Tilburg University.

Llewellyn, K. 1962. *Jurisprudence.* University of Chicago Press

MacCormick, D.N. 1981. *H.L.A. Hart*. Edward Arnold.

Marshall, C.C. 1989. Representing the structure of legal argument, *Proceedings of 2nd International Conference on Artificial Intelligence and Law.* USA: ACM Press, pp. 121–127.

Perelman, C. and Olbrechts-Tyteca, L. 1969. *The New Rhetoric*. translated by J. Wilkenson. and P. Weaver. Notre Dame, Indiana: University of Notre Dame Press. Originally published in 1958 as C. Perelman and L. Olbrechts-Tycteca 1958. *La Nouvelle Rhétorique: Traité de l'Argumentation*. Presses Universitaires de France.

Philipps, L. 1989. Are legal decision based on the application of rules or prototype recognition?: legal science on the way to neural networks, *Proceedings of the Third International Conference on Logica, Informatica, Diritto*, Vol. 2. Florence, p. 673.

Poole, D.L. 1988. A Logical framework for default reasoning, *Artificial Intelligence* **36**, 27–47.

Prakken, H. 1993. *Logical Tools for Modelling Legal Argument*. PhD thesis. Amsterdam: Vrije University.

Stranieri, A., Gawler, M., and Zeleznikow, J. 1994. Toulmin argument structures as a higher level abstraction for hybrid reasoning, in C. Zhang, J. Debenham, and D. Lukose (eds.) *Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence. AI'94.* Singapore: World Scientific, pp. 203–210.

Stranieri, A., Massey, P., and Zeleznikow, J. 1994. Inferencing with legal knowledge represented as diagrams, in A.W.F. Williams (eds.), *Poster Proceedings of the Seventh Australian Joint Con-*

ference on Artificial Intelligence AI'94. Armidale, Australia: University of New England, pp. 25–32.

Stranieri, A. and Zeleznikow, J. 1995. Levels of reasoning as the basis for a formalisation of argumentation. *Proceedings of CIKM'95, the Fourth International Conference on Information and Knowledge Management*. Baltimore, Maryland: ACM Press, pp. 333–339.

Stranieri, A, and Zeleznikow, J. 1992. Split-up: Expert system to determine spousal property distribution on litigation in the family court of Australia. In A. Adams and L. Sterling (eds.), *AI'92: Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. Hobart. Australia: World Scientific, Sydney.

Thagard, P. 1989. Explanatory coherence, *Behavioural and Brain Sciences* **12**, 435–502.

Toulmin, S. 1958. *The Uses of Arguments*. Cambridge: Cambridge University Press.

van Dijk, T.A. 1989. Relevance in logic and grammar. In J. Norman and R. Sylvan (eds.), *Directions in Relevant Logic*, pp. 25–57.

Walker, R.F., Oskamp, A., Schrickx, J.A., Opdorp, G.J., and van den Berg, P.H. 1991. PROLEXS: Creating law and order in a heterogeneous domain, *International Journal of Man Machine Studies* **35**(1), 35–68.

Warner, D. 1994. A neural network-based law machine: The problem of legitimacy, *Law, Computers & Artificial Intelligence* **2**(2), 135–147.

Weiss, S and Kulikowski, C. 1991. *Computer Systems that Learn*. California: Morgan-Kaufman.

Wick, M.J and Thompson, W.B. 1992. Reconstructive expert system explanation *Artificial Intelligence* **54**, 33–70.

Zeleznikow, J. Hunter, D., and Stranieri, A. 1997. Using cases to build intelligent decision support systems, in R. Meersman and L. Mark (eds.), *Database Applications Semantics. Proceedings of the IFIP Working Group Stone Mountain, Georgia, USA. May 30–June 2. 1995*. Chapman-Hall, pp. 443–460.