Integrating Bayesian Networks into Knowledge-Intensive CBR

Agnar Aamodt & Helge Langseth

Norwegian University of Science and Technology
Department of Computer and Information Science
N-7034 Trondheim, Norway
Agnar.Aamodt@idi.ntnu.no

Norwegian University of Science and Technology
Department of Mathematical Sciences
N-7034 Trondheim, Norway
Helge.Langseth@stat.ntnu.no

Abstract

In this paper we propose an approach to knowledge intensive CBR, where explanations are generated from a domain model consisting partly of a semantic network and partly of a Bayesian network (BN). The BN enables learning within this domain model based on the observed data. The domain model is used to focus the retrieval and reuse of past cases, as well as the indexing when learning a new case. Essentially, the BN-powered submodel works in parallel with the semantic network model to generate a statistically sound contribution to case indexing, retrieval and explanation.

1. Introduction and background

In knowledge-intensive CBR a model of general domain knowledge is utilized to support the processes of retrieval and reuse of past cases, as well as to learn new cases. The role of the general domain knowledge is to explain why two cases are similar based on semantic and pragmatic criteria, how a past solution may be adapted to a new problem case, and/or what to retain from a case just solved (Porter et. al, 1990; Branting, 1991; Aamodt, 1994; Leake, 1995). Given this master-slave relationship between CBR and some kind of model-based reasoning, a particular problem is how to develop and maintain the general domain knowledge needed. The usual way is to rely on manual knowledge acquisition. Taking the well-known problems of updating general knowledge by automated means into account, this model is often regarded as static, or only occasionally revised. The automated learning is then taken care of at the level of specific domain knowledge, i.e. the collection of cases (Aamodt, 1995). Due to this problem, reasoning from general domain knowledge is by many CBR researchers seen as counterintuitive to the very idea of CBR.

Still, problem solving and learning by combining general and case-specific knowledge seems to be what people do. Those of us who study the integrated utilization of the two knowledge types are therefore continuously looking for approaches and methods that reduce the problems just mentioned. In particular, our research addresses the problem of sustained (continuous) learning, which ensures that the system learns and correspondingly updates its knowledge after each problem solving session. Our reference model is the Creek model (Aamodt, 1994), a densely connected semantic network of prototypical concept definitions and their different relationships, in which cases are integrated. An approach that seems promising for learning of general knowledge in this type of structure is belief network learning, also referred to as Bayesian networks (BN). Significant distinctions from other learning methods include its strong statistical basis, that it maintains a symbolic representation in terms of a dependency network, and that the relations in the network may be given meaningful semantic interpretations, such as causality. BN is a heavily studied approach in the ongoing research on data mining; where the discovery of knowledge from large collections of data is addressed.

The next section describes the scope of our research within a larger perspective of data mining and CBR integration. This is followed by a summary of related research. Sections 4 and 5 describes the essence of our approach to BN and CBR integration, by explaining how BNs and semantic networks are combined in the general domain model (4), and how the BN-generated knowledge contributes to case retrieval.

2. Data mining and CBR

Within a larger scope, we are studying the combination of CBR and statistical and AI-based data mining methods. Bayesian networks provides a data mining method to the learning of a partial general knowledge model from observed data, and it utilizes already existing domain knowledge in doing so (Heckerman et al., 1995). Since the method is based on statistical treatment of data, it is a bottom-up approach complementary to the top-down approach of manual knowledge acquisition.

Our research is undertaken within the scope of the NOEMIE project (EU Esprit IV), in which data mining and CBR are combined in order to improve the transfer and reuse of industrial experience. Data mining provides a *data-oriented* view to this task, i.e. it captures potentially useful relationships based on the analysis of existing data in databases. The main role of CBR, on the other hand, is to capture, in an experience case base, past experiences that the users have had, which then represents a *user-oriented* view of the domain world. The aim of the project is to capture the knowledge structures representing these two views and develop methods that utilize them in a combined way for decision support and for information retrieval across multiple databases.

In the NOEMIE project two types of data mining methods are explored: BNs and other statistical methods such as trend analysis, clustering and factorial analysis (Casella&Berger, 1990; Diday, 1993). Bayesian networks has a primary role in maintaining the general domain knowledge, while we are investigating the possible contribution to the generation of cases from the other methods. In Figure 1, this scheme is illustrated. The arrows labeled dm(bn) and dm(other) show the primary role of BN and methods such as clustering, etc. related to the general knowledge model and the cases, respectively.

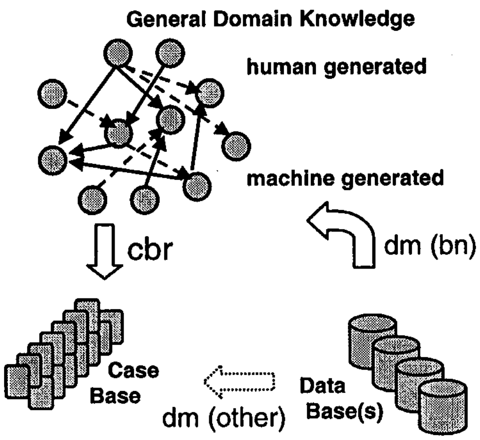

Figure 1. Knowledge/data structures and methods

The *general domain knowledge* consists of both human generated knowledge (product of manual knowledge acquisition) and knowledge generated from data by Bayesian methods. The first will typically be static knowledge such as high-level ontologies, essential textbook knowledge, and facts and heuristics commonly agreed upon among domain specialists. The latter knowledge type is dynamic in nature; i.e. suited to capture the changing aspects of a problem domain. The *case base* captures user experiences via support from the general knowledge model, but may also get support from the databases. An example of the latter is to use data mining for reconstruction of past user experiences from data in legacy databases (utilizing textual as well as formatted data).

The methods and tools resulting from the NOEMIE project will be evaluated by two pilot applications, one of which addresses problems of unwanted events in offshore drilling related to problems of lost circulation material, kicks and blowouts. The domain model contains relationships such as:

Very high mud density sometimes leads to lower fracture resistance.

A database consisting of about 1200 blowouts from the Gulf of Mexico since 1960 has been prepared (Skalle & Podio, 1998). This database will serve as a basis for the BN generated knowledge. An example:

RigType = Jacket and Outer Diameter of Casing <= 10.75 implies probability distribution of primary barrier failed equals (Swabbing 75%, Unexpected well pressure/drilling break 15%, Others 10%).

A heavily researched area within the field of Bayesian networks is to find algorithms for generating plausible network representations from data (Heckerman & al., 1995). As the problem is NP-complete, most algorithms lend themselves to some standard search method, and the problem reduces to finding a suitable search metric. The aim is to find the model with the highest probability of being correct, given the data. Through Bayes' rule, this is changed into calculating the probability of actually seeing the observed data given each model, and selecting the model with the highest score. That is, the score for a given model *M* is calculated as

$$P(\mathit{Observed\ data} \mid \mathit{Model\ M})$$

For complete datasets (i.e., no missing attributes in any data record) this calculation is straightforward. Values can be calculated locally, and each record of the database is independent. Most real-life databases are, however, not complete. Various methods are proposed for mending this problem; we use the *Bayesian Knowledge Discoverer* (Ramoni & Sebastini, 1996). The software uses the metric above to guide the search for a network representation, but is enhanced with a method termed *"bound and collapse"* to handle missing attributes in the data set.

If the domain model is properly built, (most of) the relationships found by BKD will already be in the model. In this case, the Bayesian network generated connections are only a verification of the generated model, with a quantification of the strength of each connection. Another feasible approach is to let the expert generated domain model be the *a priori* network, and use the data to generate the *a posteriori* network from this system.

The rest of this paper will focus on BN for support of general knowledge maintenance and the impact this may have on the CBR process.

3. Related research

Ongoing research, addressing master-slave combinations of CBR and BN, include (Dingsøyr, 1998):

- Microsoft Research has developed a three-layer Bayesian network for fault diagnosis (Breese & Heckerman, 1995) called Aladdin. The network, which is constructed by an expert, represents the cases in the case base. The BN is updated each time a new case is seen.
- The University of Helsinki has developed D-SIDE, a tool designed for data-intensive domains (Tirri et. al.,

1996). Cases are viewed as multi-dimensional stochastic vectors where input noise may be present. During *Retrieve* they use a Bayesian network to select the most similar case. Missing feature values are filled in using the MAP estimator.

- Aha & Chang (1996) uses a Bayesian network to select a maximum feasible plan in multiagent soccer, given the current situation. CBR is used to retrieve a case with the plan implemented. This paper also gives a review of similar research.
- At the University of Salford, two BNs are used in an exemplar-based CBR system (Rodriguez et. al., 1997). The first is used to index categories of cases with some similarity, the other to index cases within each group of similar cases.

In Table 1 these systems are compared with respect to their role in the four main CBR processes (extended from (Dingsøyr, 1998)).

Table 1: Related systems

System	*MS Research*	*Aha&Chang*	*Helsinki*	*Salford*
Retrieve	Use BN to select case	Use BN to select category of cases	Use BN to select case[1)]	Identify case in BN
Reuse	Not addressed	Independent of the BN	Not addressed[1)]	Not addressed
Revise	Not addressed	Independent of the BN	Not addressed	Not addressed
Retain	Update BN (add case)	Update BN, add case	Not addressed	Index cases using BN

[1)] The Helsinki group uses the term "reuse" for the sub-process, which we name "retrieve" in our terminology.

Additional activities: Chang & Harrison (1995) use several data mining techniques including Bayesian networks to assist the CBR process. Daphne Koller and colleagues (Koller et. al., 1997; Boutlier et. al., 1996) are working on Bayesian networks as a functional programming language in stochastic domains. Their work includes machinery for object oriented Bayesian networks and a method to extract context specific acyclic subgraphs for specific inference tasks.

Our proposed system uses a BN to select a set of cases during *Retrieve*. *Reuse* and *Revise* are independent of the BN. During *Retain* the BN is updated and the case is added. Hence, Microsoft Research's Aladdin and Salford system are most similar to our approach. The main difference between our method and the two cited is in the role of the cases. Opposed to Aladdin our system makes explicit use of the cases in the user interaction. In Aladdin, each user query is only answered with the results from a BN inference task. The system generated at Salford also uses a more shallow case definition than our.

4. Combining Bayesian and semantic networks

Bayesian networks are constructed using the vast amount of background data stored within the company. Our goal is to generate a Bayesian network containing the *observable* attributes of the domain model both to discover new relationships between them, and to verify the dependencies already entered by the domain modeler. The result is a submodel of statistical relationships, which will live their own life in parallel with the "classic" model. The BN generated submodel is *dynamic* in nature i.e. we will continuously update the strengths of the dependencies as new data is seen. This is opposed to the static relationships of the classic domain model. The dynamic model suffers from its less complete structure (we will only include observable terms in this model) but has an advantage through its sound statistic foundation and its dynamic nature. Hence, we view the domain model as an integration of two parts:

- A "static" piece consisting of relations assumed not – or seldom - to change (like has-subclass, has-component, has-subprocess, has-function, always-causes, etc).
- A "dynamic" part, which is made up of dependencies of a stochastic nature. In changing environments, the strengths of these relations are expected to change over time. (Example: causes - in a more general sense)

A semantic network handles the static part. The Creek system (Aamodt, 1994) for explanation-driven CBR represents general domain knowledge as a densely coupled network of concepts and relations. Figure 2 gives a visual impression of a part of such a network in the domain of car starting problems.

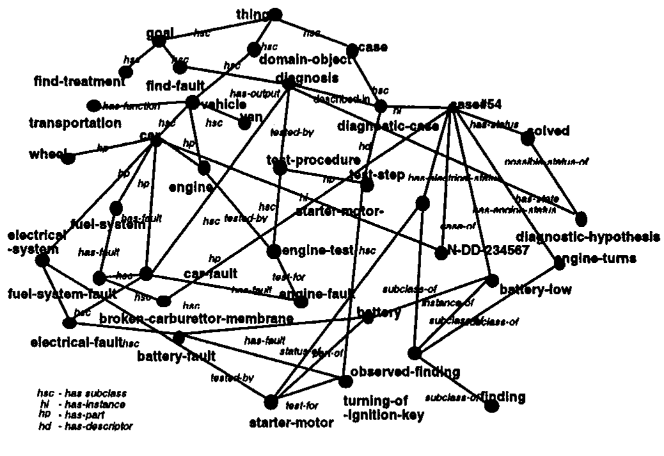

Figure 2: A tangled network of CreekL concepts

Each node and each link in the network is represented as a frame object. As shown, a case is also a node in the network, linked into the rest of the network by its case features. Below is a frame showing a representation of a solved case, described by its feature names (slot names) and values.

case#54

instance-of	value	car-starting-case diagnostic-case
has-task	value	find-car-starting-fault
has-status	value	solved
of-car	value	N-DD-234567
has-fault	value	broken-carburetor-membrane
has-fault-explanation	value	(.....)
has-repair	value	replace-carburettor-membrane
has-electrical-status	value	battery-low starter-motor-turns
has-engine-status	value	engine-turns engine-does-not fire
has-ignition-status	value	spark-plugs-ok
has-weather-condition	value	low-temperature sunny
has-driving-history	value	hard-driving

The has-fault-explanation slot (left out for clarity reasons) contains an explanation of the fault as a path – or set of paths – in the domain model, linking the fault to the observed findings (see below).

The dynamic portion of the domain model extends the total model with a new relation, which we name bn-influences. This relationship is learned from the data through a data mining process as described above. BN inference methods are subsequently used to make similarity calculations by following the corresponding links in the domain model. The result is an additional calculation method to compute similarities, generate indexes, and guide the user when entering information. Supplementary explanations can also be generated (Chajewska & Halpern, 1997).

Empirical results from using BNs in different domains indicate that classification is surprisingly robust when it comes to errors in both network structure and local distribution functions (Heckerman et al. 1995, Henrion et al, 1996). Hence, the case retrieval process (which is an inference task in our setup) is assumed stable.

Note that the set of vertices building up the Bayesian network consists of more than the dynamic domain model. We include a superstructure consisting of vertices necessary to make sufficiently advanced inference in the Bayesian network. Our proposed algorithm is as follows:

- Include all observable nodes. These can be *symptoms* (leaf nodes) or *influence factors* ("top nodes")
- Include all *target nodes*, i.e. nodes which hold the (concealed) information we are searching for
- Any node on an *influence path* from the influence factors via the target nodes to the leaf nodes.
- A superstructure is made when two or more siblings (vertices with a common parent in the static domain model) have strong similarities, but are not connected in the dynamic model. We then add the parent to the Bayesian network vertices. As an example, fuel-system-fault is included in the dynamic model (see Figure 3), because the analyst felt there is a resemblance between carburetor-fault and water-in-gas-tank.
- Cases are represented as binary variables, with their relevance given as the probability of being "On". Cases are *leaf nodes* (i.e. they have no children), and their parents are given by the set of relevant findings for this case.

In Figure 3 a causal explanation structure for the car domain, derived from the model illustrated in Figure 1, is shown. A simplified Bayesian knowledge structure for the same domain was found by following the steps described above. The result is superimposed in the picture. Vertices that are both in the Bayesian and "classic" domain models are black; vertices only in the classic model are white.

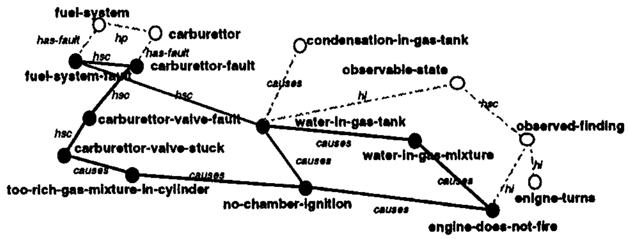

Figure 3: Partial knowledge structure from the car domain. The Bayesian network is superimposed

Edges included in both the Bayesian network and the classic domain model (these are the relation types hsc and causes for this simple example) are black; edges only in the classic domain model are dotted. Hence, the causes relations along full, black lines should be interpreted as causes + bn-influences. The network only identifies which nodes that are part of the BN. Since no data has been seen yet, there are no conditional dependency tables attached to the bn-influences links.

The network structure predefined in this manner may in principle also be altered by the data mining part. The dynamic domain model is incrementally monitored and updated as new data is given to the system. This Bayesian learning, which is orthogonal to learning through storage of new cases, consists of two parts; both the qualitative structure of the network and the strength of each edge are adjusted. This incremental learning of the edge strengths is straightforward using standard methods for complete data sets; when the data is prone to missing values, the EM algorithm (Dempster & al., 1977) or the steepest ascent (Binder & al., 1997) must be used. Methods to incrementally control the model structure exist as well. (Lam & Bacchus, 1994) propose a method based on the MDL principle, simpler methods (like peeling off Noisy-OR gates) are also applicable. A heavily researched task in the BN community is to define theory to detect and handle *hidden variables* (Heckerman 1997). One tries to pinpoint unknown conditions (i.e., not in the domain model) influencing the domain by looking for variations in the observed behavior not explained by the domain model. In our approach, these methods will serve as a technique to find sub models of the domain that are poorly modeled.

5. Bayesian case retrieval

Each case is indexed by the Bayesian network by a binary feature link (on or off, with probability). The standard Creek process of choosing index features is adopted, taking both the predictive strength and necessity of a feature into account. The retrieval of cases is a two step process:

Pass 1: The observed features (from the new case) are entered as evidence into the Bayesian network. The

Bayesian network is updated (i.e., the inference method is applied), and each case is allocated a probability of being *On*. This is calculated on the basis of the conditional probability *P(Case node is On|Features of New case)*, which then represents the similarity metric. Every case with a probability exceeding some threshold value, *t*, is retrieved. The context specific value *t* must be found through trial and error, typically between 5 and 10 cases are activated in Pass 1.

The procedure is exemplified in Figure 4. One feature node is instantiated (drawn black), and the information flows through the nodes in the network. The affected nodes ("vertices" in BN terminology) are shaded according to the degree of influence. Eventually, the information flows to the case nodes (drawn as squares in the network). One of the cases are affected by the information entered (i.e., its probability for being "On" is altered), the other is not affected by the information entered.

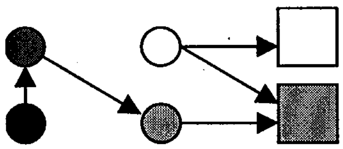

Figure 4: Retrieve in the Bayesian network

Pass 2: The cases that were activated during Pass 1 are retrieved for further processing. We want to select $k \geq 1$ of the plausible cases, and to do so the system infers what type of information that is best suited for discriminating between the plausible cases. A BN approach will be to look up one of the (yet unseen) observable nodes. If, in Figure 4, both cases shown are retrieved, a choice for the discriminating information may be the node that influences both cases. Pass 2 is repeated until *k* cases are standing out as the most plausible.

This approach can be combined with the explanatory process of a basic Creek system (Aamodt, 1995), either as two separate methods or as a combined method. We are investigating both.

6. Discussion

We are currently in the process of implementing the BN method into the Creek system in order to test out the different integration options. The main CBR engine in the NOEMIE project is the KATE system, which now is being extended and linked to Creek methods. The methods will be adapted to incorporate the use of BN that turns out most successful.

A particular feature of the Bayesian network submodel is its dynamic nature i.e. the strengths of the dependencies are continuously updated as new data is seen. Hence, the system is adaptable, both because new cases are stored in the system, but also because we are able to continuously monitor the adequacy of the domain model. The main disadvantage is the extra computational complexity introduced; hence domains which are static and deterministic, or without an interesting causal structure (like recognition of faces) are not suited for the proposed integration.

The dynamic BN submodel suffers from its less complete structure, but has an advantage through its sound statistic foundation. Therefore the BN submodel cannot replace the classic domain model; the two must work in cooperation.

References

Aamodt, A. 1994: Explanation-driven case-based reasoning, In S. Wess, K. Althoff, M. Richter (eds.): Topics in Case-based reasoning. Springer Verlag, pp 274-288.

Aamodt, A. 1995: Knowledge Acquisition and Learning from Experience - The Role of Case-Specific Knowledge, In Gheorge Tecuci and Yves Kodratoff (eds): Machine learning and knowledge acquisition; Integrated approaches, (Chapter 8), Academic Press, pp 197-245.

Aha, D. W & Chang, L. 1996: Cooperative Bayesian and Case-Based Reasoning for Solving Multiagent Planning Tasks, Technical Report, AIC-96-005, Navy Center for Applied Research in Artificial Intelligence

Binder, J., D. Koller, S. Russell, K. Kanazawka 1997: Adaptive Probabilistic Networks with Hidden Variables *Machine Learning*, in press,

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

Breese, J. & Heckerman, D, 1995: Decision-theoretic case-based reasoning. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 56-63, 1995.

Boutlier, C. Friedman, N., Goldszmidt, M. and D. Koller, 1996 Context-Specific Independence in Bayesian Networks, In Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, 115-123

Casella, G. & Berger, R.L., 1990: Statistical Inference Wadsworth Inc., California.

Chang, L & Harrison, P. 1996. A Testbed for Experiments in Adaptive Memory Retrieval and Indexing, Proceedings of AAAI95 fall symposium.

Chajewska, U & J. Halpern, 1997, Defining Explanations in Probabilistic Systems, In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, California, 62-71.

Dempster, A., Laird, N., and Rubin, D. 1977: Maximum Likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*. Series B, pp. 1-38.

Diday, E. 1993: An introduction to symbolic data analysis INRI, No 1936, Aout 1993

Heckerman, D. (1997): Bayesian Networks for Data Mining, *Data Mining and Knowledge Discovery 1*: 79-119

Heckerman, D., D. Geiger and M. Chickering (1995) Learning Bayesian networks, the combination of knowledge and statistical data, *Machine Learning* 20: 197-243.

Henrion, M., Pradhan, M., Del Favero, B., Huang, K., Provan, G., O'Rorke, P. 1996: Why is diagnosis using belief networks insensitive to imprecision in probabilities? In Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, California, 307-314.

Koller, D. and A. Pfeffer, 1997: Object-Oriented Bayesian Networks, In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, California, 302-313.

Lam, W. & F. Bacchus, 1994: Using New Data to Refine a Bayesian Network, In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, California, 383-390.

Leake, D. 1995: Combining rules and cases to learn case adaptation. In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society, pp 84-89. Ann Arbor, Mich. Cognitive Science Society.

Ozturk, P. & Aamodt, A. 1997: Towards a model of context for case-based diagnostic problem solving. In Context-97; Proceedings of the interdisciplinary conference on modeling and using context. Rio de Janeiro, February 1997, pp 198-208.

Ramoni, M. and P. Sebastiani (1996): Learning Bayesian Networks from Incomplete Databases. Technical report KMI-TR-43, Knowledge Media Institute, The Open University.

Rodriguez, A. F., S. Vadera, and L. Enrique Sucar 1997: A probabilistic model for case-based reasoning. In I Smith and B Faltings, editors, *Case-Based Reasoning Research and Development, ICCBR-97. Proceedings*, pages 623-632.

Skalle, P. & Podio, A. L. 1998: Trends extracted from 1200 Gulf Coast blowouts during 1960-1996. AIDC/SPE Drilling Conference, Dallas, USA 3-6 March 1998.

Tirri, H., P. Kontkanen and P. Myllymäksi 1996: A Bayesian framework for case-based reasoning. In I Smith and B Faltings, editors, *Advances in Case-Based Reasoning, EWCBR-96*, pages 413-427.

Appendix A

1. **Integration name/category**: Bayesian Networks and CBR.

2. **Performance Task**: Any, but primarily diagnosis and information focusing in open, weak theory domains.

3. **Integration Objective**: Bayesian network (BN) assists in retrieval and indexing, and in learning of general domain knowledge.

4. **Reasoning Components**: BN (for computing similarity) and maintaining a "dynamic" domain model.

5. **Control Architecture**: Interleaved (BN used within subprocesses of CBR).

6. **CBR Cycle Step(s) Supported**: Retrieval, Reuse, and Retention.

7. **Representations**: Semantic net for domain knowledge, the BN is a subset of semantic net with additional conditional probabilities. In the BN, case nodes are binary variables linked to the case features.

8. **Additional Reasoning Components**: Model-based reasoning.

9. **Integration Status**: Proposal building on previous work (Creek).

10. **Priority future work**: Empirical evaluation.