A Bayesian Approach to Automating Argumentation

Richard McConachy, Kevin B. Korb & Ingrid Zukerman
Department of Computer Science, Monash University
Clayton, Victoria 3168, AUSTRALIA
{ricky,korb,ingrid}@cs.monash.edu.au

Abstract

Our argumentation system NAG uses Bayesian networks in a user model and in a normative model to assemble and assess nice arguments, that is arguments which balance persuasiveness with normative correctness. Attentional focus is simulated in both models to select relevant subnetworks for Bayesian propagation. Bayesian propagation in the user model is modified to represent some human cognitive weaknesses. The subnetworks are expanded in an iterative abductive process until argumentative goals are achieved in both models, when the argument is presented to the user.

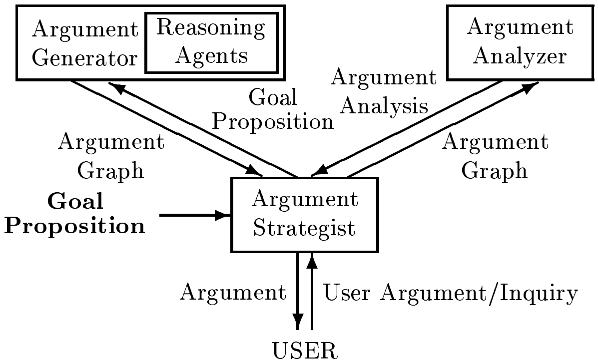

Figure 1: System Architecture

1 Introduction

"The St. Kilda Football Club will probably win the grand final in 1997. They are likely to do so because they have won at least as many games as any other side during the regular season, they have scored more points than any other team this year so far, and they have few unavailable players." How strongly should St. Kilda's good form in the later part of the 1997 season be taken as a premiership indicator? Is it better to lose narrowly to a strong side or narrowly defeat a weak side? Most arguments, like the St. Kilda example above, must be formed using information that is both incomplete and uncertain.

This paper focuses upon the Bayesian underpinnings of our argument generation-analysis system, NAG (*Nice Argument Generator*). The goal of our system is to generate *nice* arguments. A nice argument employs normatively strong inferences from accepted premises while also being persuasive for the target audience.

The main modules of NAG are shown in Figure 1. The Generator (Section 3) uses semantic activation to quickly form the initial *Argument Graph* for an argument, or to quickly extend an already existing Argument Graph. An Argument Graph is a network structure with nodes that represent propositions, and connecting links that represent the inferences that connect these propositions. An Argument Graph is fleshed out by consulting several sources of information called *Reasoning Agents* and incorporating the relevant inferences and propositions returned by these sources into the Argument Graph. This Argument Graph is passed to the Strategist.

The Strategist decides what NAG should do next: call the Generator to continue the argument building process; call the Analyzer (Section 4) to estimate how nice the current Argument Graph is; or present an argument based on the current Argument Graph to the user for inspection and response.

The Strategist will pass the current Argument Graph to the Analyzer at least once before the argument is presented to the user, and often more than once. To estimate the persuasive power of an argument represented by an Argument Graph, the Analyzer consults a revisable user model that reflects the beliefs and cognitive abilities of the audience. The Analyzer uses a normative model to gauge the normative strength of an argument. Belief updating in both the user and the normative model is done by a constrained Bayesian propagation scheme (Sec-

Richard McConachy, Kevin B. Korb and Ingrid Zukerman (1998) **A Bayesian Approach to Automating Argumentation.** In D.M.W. Powers (ed.) *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning,* ACL, pp 91-100.

tion 4). In the user model, Bayesian updating is adjusted by multiplicative factors which model three human cognitive weaknesses (Section 4.1).

If the Analyzer detects problems with the Argument Graph it highlights the weaknesses for the Generator to fix. In this way a cycle of alternately critiquing and extending the graph is continued until a successful Argument Graph is built, or NAG is confronted with an event that prevents it from continuing, such as the Generator failing to find relevant new evidence or the Strategist noticing that the allowed amount of time has run out.

2 Related Research

Charniak and Goldman (1993) describe a Bayesian plan recognition system that uses marker passing as a method of focusing attention on a manageable portion of the space of all possible plans. This is similar to the way in which NAG uses spreading activation to focus on a small portion of the available data when generating arguments.

The approach of "interpretation as abduction" used in (Hobbs *et al.*, 1993) aims to recover the premises and inferential links which lead to the conclusion of some given argument. This is similar to NAG's argument analysis. There are two important differences between NAG and the work by Hobbs *et al.*: NAG is a system that reasons under uncertainty, and NAG performs both analysis and generation. A generative system based on the work of Hobbs *et al.* is described in (Thomason *et al.*, 1996). This system deals with what can be readily inferred, and so deleted, during communication, but the generated discourse does not present an argument in support of a proposition. Two systems that can turn an existing fully explicit argument into an enthymematic one are described in (Horacek, 1994) and (Mehl, 1994). However, both of these systems require a complete argument as input, unlike NAG, which constructs its own.

3 Argument Generation

The Generator receives the following inputs: (1) a proposition to be argued for; (2) an initial argument context; (3) two target ranges of degrees of belief to be achieved (one each for the normative model and the user model); and (4) a system attitude parameter, which determines the extent to which the system will take advantage of the user's erroneous beliefs.[1] An argument context is the set of propositions and concepts appearing in the discussion preceding the argument or in the current Argument Graph. The degrees of belief to be achieved are expressed as ranges of probabilities, e.g., [0.5, 0.6], in order to be able to represent a variety of goals, e.g., inducing indifference or assent.

The Generator produces an Argument Graph which starts from *admissible premises* and ends in the goal proposition. Admissible premises are normatively acceptable propositions that are either already believed by NAG and the user (sufficiently for the argument to work) or are assented to by the user (e.g., sourced from an accepted reference work). When constructing an argument, the Generator relies on two collections of information: a normative model composed of different types of *Knowledge Bases* (KBs) and a user model also composed of different types of KBs which represent the user's presumed beliefs and inferences. A single KB represents information in one format, e.g., a semantic network (SN), Bayesian network (BN), rule-based system, or database. During the argument generation process, relevant material from several of the KBs may need to be combined into a common representation. We have chosen BNs for this purpose because of their ability to represent normatively correct reasoning under uncertainty, and because altering the normal Bayesian propagation rules allows us to model some human cognitive phenomena.

When assembling an Argument Graph, NAG develops two BNs: the BN forming one of the KBs in the user model, and the BN forming one of the KBs in the normative model. As arguments are built up, material obtained from other KBs may be converted to BN form and added to the appropriate BN, e.g., material from a rule-based system in the normative model may be added to the normative BN (Section 3.4). In order to reduce the amount of information NAG must deal with, a focusing mechanism is applied to highlight the portion of the complete BN in each model needed for the current argument (Section 3.2). Hence, both the user model and the normative model will contain a single Bayesian subnetwork that is under focus. The structural intersection of these Bayesian subnetworks form the Argument Graph. When analyzing the current Argument Graph, propagation is performed twice, once over the Bayesian subnetwork in the user model and once over the Bayesian subnetwork in the normative mode, each time using probabilistic information sourced from within the model being propagated (Section 4). Thus, we obtain the strength of the same argument in both the user and normative models.

[1]The results in this paper are for a "normal" attitude, which allows a small departure from what is normatively correct.

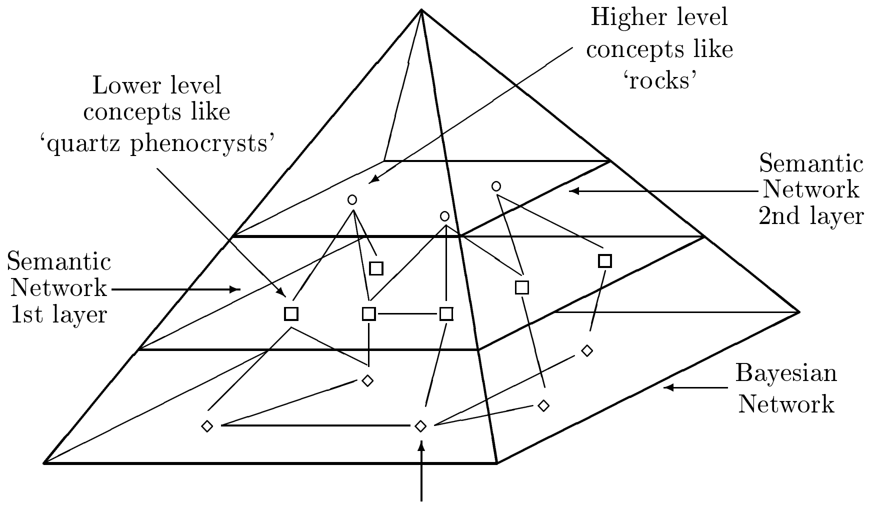

Figure 2: Semantic and Bayesian Networks

3.1 The Generation-Analysis Algorithm

As previously described, part of NAG's input is an initial argument context, $context_0$, which is defined by the propositions and concepts mentioned in the preamble to the argument plus the argument's goal. Having established the initial context, NAG activates the following algorithm:

Generation-Analysis Algorithm

1. $i \leftarrow 0$.
2. Clamp any items related to the current context, $context_i$, and perform spreading activation. This results in an Argument Graph containing: the clamped nodes, the activated nodes (those whose activation exceeds a threshold), plus the links connecting the nodes. **(Section 3.2)**
3. Identify new subgoals in the current Argument Graph and tag them for further investigation. **(Section 3.3)**
4. Pass each of the argument subgoals identified in the previous step to the Reasoning Agents, and add the new information returned by the Reasoning Agents to the current Argument Graph. **(Section 3.4)**
5. Pass the current Argument Graph to the Analyzer for evaluation. **(Section 4)**
6. If the Analyzer reports that the current Argument Graph is sufficiently nice, then present an argument based on the current Argument Graph to the user, and wait for a response. **(Section 5)**
7. $i \leftarrow i + 1$.
8. $context_i \leftarrow context_{i-1}$ + *new nodes connected to the goal during the cycle* i–1.
9. Go to Step 2.

3.2 Focusing the Argument

Bayesian network propagation (Pearl, 1988), optimized or otherwise (see, for example, Li & D'Ambrosio, 1994), is an NP-hard problem in the general case (Cooper, 1990). NAG is designed to be an interactive system, potentially drawing upon very large knowledge bases, so complete propagation over large BNs would be too slow. In addition, NAG's user model is designed to model human cognitive abilities, and humans normally cannot absorb and analyze all data relevant to a complex problem. To cope with both of these limits on complexity we emulate the principal means available to humans for applying limited cognitive capacity to problem solving, namely attention (see, for example, Baars, 1987).

NAG uses two hierarchical SNs, one built on top of the user model BN and one built on top of the normative model BN, to capture connections between the items mentioned in the discourse. Figure 2 illustrates one such semantic-Bayesian 'pyramid'. The SN (upper levels of the pyramid) and the BN (base of the pyramid) are used by NAG to simulate attentional focus in each model. NAG takes the context in which the argument occurs as providing an initial set of salient objects. For example, if the user presents an argument to NAG, the concepts occurring in the propositions within the argument or in the preceding discussion will be marked as salient. We use activation with decay (Anderson, 1983), spreading from the salient objects (which are clamped) to determine the focus of attention. All items in the semantic-Bayesian networks which achieve a threshold activation level during the spreading activation process are brought into the span of attention. The spreading activation process passes activation through the pyramidal semantic-Bayesian networks, each node being activated to the degree implied by the activation levels of its neighbors, the strength of association to those neighbors, and its immediately prior activation level (vitiated by a time-decay factor). By these means we have a direct implementation of attention which we use to zero-in upon the more useful portions of the pyramidal semantic-Bayesian networks. This iterative process ceases when an activation cycle fails to activate any new node.

Determining the more useful portions of the pyramidal semantic-Bayesian networks in this way allows NAG to save processing time in two ways: NAG can restrict itself to searching for information connected with only the most relevant propositions (the ones in focus, see Section 3.3), rather than all of the propositions known to the system; and similarly, NAG can analyze its arguments with respect to just the same relevant propositions, saving time in the Bayesian propagation procedure (Section 4).

3.2.1 Focusing Example

Consider the generation of an argument for the proposition "A *porphyry copper deposit* exists at *Mineral Park, Arizona,*" preceded by the preamble "A *porphyry copper deposit* derives its name from a *porphyritic stock* located at the center of the *mineral deposit.* A *stock* is a cylindrical *igneous* mass working its way up through the earth's crust. The word *porphyritic* describes the bimodal nature of the *minerals* within the *stock:* some of these *minerals* are large and the rest are microscopic. For example, *porphyritic granite* at *Mineral Park* usually contains large crystals *(phenocrysts)* of *feldspar* and *small grains* of *quartz.*" (this example is adapted from Duda *et al.*, 1978). Initially, the goal proposition and the preamble activate any propositions containing one or more of the italicized concepts, i.e., nodes N_1–N_4, N_6–N_8, and N_{13} in Figure 3 (shown in grey boxes).

After clamping the nodes that correspond to this discourse context and performing spreading activation, additional nodes become activated in the semantic-Bayesian networks. All nodes whose activation level exceeds a threshold are retained and added to the Argument Graph. For this example, this yields an Argument Graph composed of two argument fragments: (1) N_1–N_3, N_5, N_6, N_8 and N_{11}–N_{13}; and (2) N_4, N_7 and N_{10} (Figure 3).

3.3 Choosing Argument Subgoals

Having used the semantic priming process to add items of likely interest to the current Argument Graph, NAG must now decide which of these newly added items should be set as argument subgoals requiring further inspection. At present, all nodes connected to the goal in the current Argument Graph that have not been previously passed to the Reasoning Agents are tagged as subgoals to be investigated (Section 3.4).

We are currently looking at methods for reducing the list of current subgoals. One technique for achieving this is to choose a subset of the current list of subgoals that seems to offer good potential for adding new support to the overall argument goal with minimal additional effort. A node is deemed to have a high potential for support if changes in its value would cause a large change in the final probability of the goal node after propagation (Section 4), and the node has not previously been passed to the Reasoning Agents (in which case NAG would have already examined the potential contribution of this node). This scheme of selecting only some of the available options for further inspection by the Reasoning Agents has not yet been implemented.

3.3.1 Subgoals – Example Continued

Since none of the nodes in the current Argument Graph (Figure 3) have been passed to the Reasoning Agents at this point, all of the nodes connected to the goal in the Argument Graph are tagged as subgoals. In this example, the nodes N_1–N_3, N_5, N_6, N_8, N_{11} and N_{12} are tagged as subgoals, and together with the goal node, N_{13}, are passed to the Reasoning Agents in order to obtain additional information (Section 3.4). The nodes that form the second fragment shown in Figure 3 – nodes N_4, N_7 and N_{10} – are kept, but are not expanded upon at this stage.

3.4 Extending the Argument Graph

The initial Argument Graph consists of the subset of the BNs which was activated by the attentional mechanism. The Generator then activates the Reasoning Agents to collect information relevant to each subgoal in the current Argument Graph. During the process of adding this new information to the Argument Graph, the Generator must determine: (1) which of the newly returned inferences should be integrated into the Argument Graph; (2) the struc-

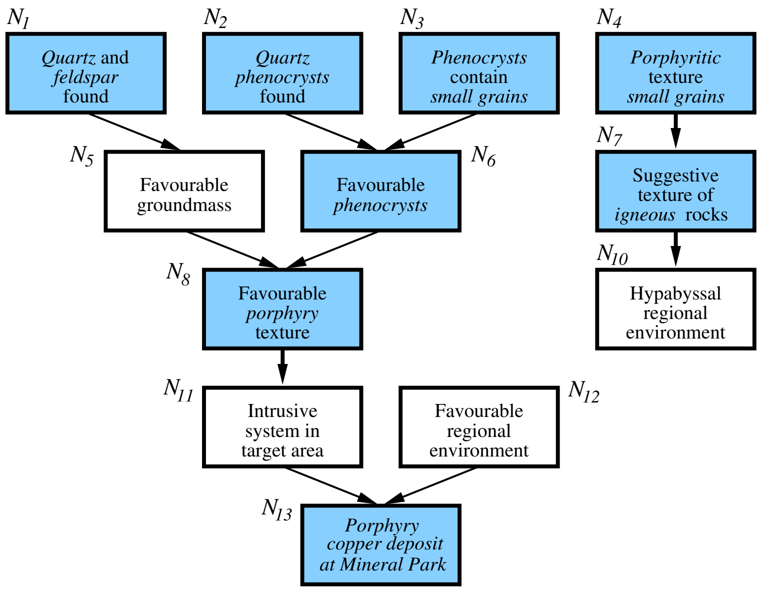

Figure 3: Initial Argument Graph for the Porphyry Copper Deposit Example

ture of the additions to the Argument Graph representing the new inferences; and (3) the parameters of the new inferences and propositions.

Which propositions and inferences to integrate. New propositions returned by the Reasoning Agents are added to the current Argument Graph as new nodes. NAG decides whether to introduce new inferences returned by the Reasoning Agents into the Argument Graph (or to replace existing inferences with new ones) by applying two simple rules designed to ensure that each relationship between propositions in the Argument Graph is represented only once. The two rules state that:

1. *At most only one inference may directly connect any two propositions in the current Bayesian subnetworks in each of the user model and the normative model.*

2. *When selecting from multiple candidate inferences, preference is given to inferences sourced from more expressive representations, where expressiveness means how much probabilistic information, including joint interactions with other factors, can be expressed by the representation.*

For example, assume NAG's qualitative rule-based system agent finds a rule stating "If D then E is possible." If the agent responsible for quantitative rule-based systems also finds a rule "If D then E with prob $= x$," which NAG translates into $D \stackrel{evidence}{\longrightarrow} E$ with $P(E|D) = x$ (assuming independence from other links incident upon node E), then which of these inference rules, if any, should be added into the Argument Graph? The first rule above states that at most one of these two inferences will be incorporated into the current Argument Graph.[2] NAG selects which one of the two inferences it will incorporate by applying the second rule. NAG implements the second rule via the following preference ordering for expressiveness: BNs, quantitative rule-based systems, qualitative rule-based systems and finally database lookups. The expectation is that the representations higher in the list, e.g., BNs and quantitative rule-based systems, will usually provide more complete and accurate probabilistic information about an inference or proposition than representations lower in the list, e.g., qualitative rule-based systems. Sometimes this expectation proves false. For example, if both a quantitative rule-based system and a BN return a probability estimate for a particular proposition being true, it is possible that the value returned by the rule-based system will be the more accurate one, perhaps because the BN has not been fully propagated (and hence the probability for the proposition not updated to incorporate all of the available information). However, initial testing suggests that NAG can often do a good enough job, i.e., produce nice enough arguments, using our two simple information lossy rules.

Structural form of the new propositions and inferences. The various Reasoning Agents return

[2]NAG does not try to merge information gleaned from more than one available source since it is unclear how to do so.

argument fragments which take the form of propositions linked by inferences and semantic relations. After the two rules mentioned previously in this section have been applied to determine which of these fragments should be incorporated in the Argument Graph, the selected fragments are added to the Bayesian subnetwork in the appropriate model, e.g., fragments sourced from KBs in the normative model will be added to the subnetwork in the normative model. Inferences in the fragment which are going to replace inferences already in the subnetwork (the new inference is from a more expressive KB than the one currently in the subnetwork, see rule two described previously) are handled in the manner described below.

Adding parameters for the propositions and inferences. Incorporating new information about the prior probability of a proposition into the Argument Graph requires that the information returned by a Reasoning Agent be copied to the node in the Argument Graph which corresponds to the KB data point from which the information was sourced. For example, information about priors sourced from a KB in the user model is copied to the corresponding node in the Bayesian subnetwork belonging to the user model. Similarly, information about possible instantiation values for a proposition are also copied to the appropriate proposition in the correct model, e.g., [Fred is a smoker] may be instantiated as TRUE (1.0) or FALSE (0.0). This works so long as the new values are filling gaps in the Argument Graph. However, if the current Argument Graph already contains a prior probability or instantiation value for the proposition under consideration, then that previous value will be retained and the new information ignored.

Adding information to the Argument Graph about joint conditional probabilities associated with new inferences is a harder task. If a Reasoning Agent can provide complete conditional probability information for a new inference which takes into account other inferences that impinge upon the proposition targeted by this inference, then this information is added to the conditional probability matrix for the node containing this proposition in the Argument Graph. However, complete probabilistic information is frequently not available to NAG, in which case the new information (often a simple conditional probability) is assumed to be conditionally independent of the other inferences impinging upon the node in question. Since assuming conditional independence without a good understanding of the domain is dangerous, NAG reports this assumption to its operator interface. This is done so that a human operator can, then or later, spot where NAG is most likely to have "gone wrong" should one of its arguments appear incorrect. The operator can then edit NAG's KBs to remove the offending inference or to add extra information about the joint conditional probabilities.

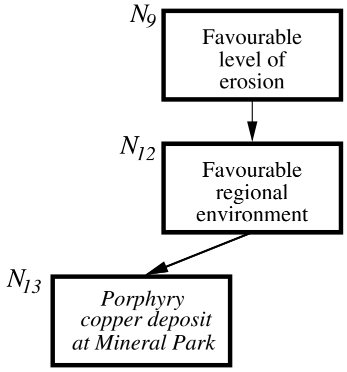

Figure 4: Results of node N_{12} being sent to the Reasoning Agents

3.4.1 Extending the Argument Graph – Example Continued

The information returned by the Reasoning Agents is incorporated into the Argument Graph. Some of this information will be included in the final Argument Graph presented to the user, e.g., the newly found node N_9 and the link connecting $N_9 \rightarrow N_{12}$ shown in Figure 4 (nodes N_{12} and N_{13} and the link between them were already contained in the current Argument graph). Other fragments may also be created during this process, e.g., [presence of igneous rocks] $\stackrel{evidence}{\longrightarrow}$ [history of high temperature and pressure conditions], but are eventually excluded because of their irrelevance to the goal.

Node N_9 and the link $N_9 \rightarrow N_{12}$ were returned when the Reasoning Agents were passed node N_{12} for inspection (recall that node N_{12}, like all of the nodes in the left hand fragment of Figure 3, was previously set as a subgoal to be investigated further). In this example, node N_{12} is the only subgoal shown in Figure 3 about which NAG could find new information.

4 Argument Analysis

The process of computing the anticipated belief in a goal proposition as a result of presenting an argument starts with the belief in the premises of the Argument Graph and ends with a new degree of belief in the goal proposition. The Analyzer computes the new belief in a proposition by combining the previous belief in it with the result of applying the inferences which precede this proposition in the

Argument Graph. This belief computation process is performed by applying propagation procedures to the Bayesian subnetwork corresponding to the current Argument Graph in the user model and separately to the subnetwork corresponding to the current Argument Graph in the normative model.

In propagating only over the subnetworks initially seeded by the focusing mechanism (Section 3.2) and extended with information returned by the Reasoning Agents (Section 3.4), NAG is ignoring those parts of the complete BNs in the user and normative models not deemed relevant to the current argument. Propagating over the subnetwork corresponding to the current Argument Graph in each of the user model and the normative model is much faster than having to perform propagation over the complete BN in each model, but the trade off is a less accurate estimate of the final belief in the goal proposition. Nonetheless, in a system designed to be interactive, some such trade off is necessary in view of the complexity of Bayesian propagation.

After propagation, the Analyzer returns two measures of an argument's strength: *normative strength*, which is the effect of an argument on the belief in the goal proposition in the normative model, and *effectiveness*, which is its effect on the user's belief in the goal proposition. Of course, the effectiveness of an argument presented to a user may be quite different from its normative strength. When determining an argument's anticipated effect upon a user, NAG currently takes into account three cognitive errors that humans frequently succumb to: belief bias, overconfidence and the failure to use base rate information (Section 4.1).

After the Analyzer has evaluated the normative strength and effectiveness of the Argument Graph it will return an assessment. This assessment may point out flaws in the argument structure such as insufficient support for some propositions within the Argument Graph and reasoning loops. At present flaws are fixed in the order they are found. The determination of a preferred ordering in which to address flaws in an argument is left for future research.

In the case of an insufficiently supported premise, the target premise is set as a new argument subgoal. This new subgoal is then submitted to the Generator with the currently active subnetworks as its context, in order to generate a new subargument. After integrating the new subargument into the previous subnetworks, the now enlarged Argument Graph is again sent to the Analyzer for inspection. Hence by allowing NAG to complete additional focusing-generation-analysis cycles, Argument Graphs that are initially unsatisfactory can often be improved. In more extreme cases where additional cycles do not gather any new evidence, altering the parameters of the semantic priming procedure, e.g., increasing the initial clamping value or decreasing the rate of decay, may help NAG identify a new subgoal that can be profitably examined by the Reasoning Agents.

4.1 Modeling Human Weaknesses

During belief updating in the user model, multiplicative factors are incorporated into the Bayesian update formulas (Neapolitan, 1990) to model the human cognitive weaknesses of belief bias, overconfidence and the base rate fallacy (for a fuller description see Korb *et al.*, 1997). In future we shall model additional prominent features of human inference.

Belief bias is the assessment of an inference as being stronger (weaker) than it is normatively because it supports (undermines) an existing belief (Evans, 1989). To model this effect we employ a function which given the user's prior degree of belief in a proposition provides a multiplicative factor used in any update to that belief. Should the user's prior belief be 0.5, then the multiplicative factor is 1, so that the belief is updated normally. An extreme prior disbelief on the other hand suppresses the impact of supporting evidence via a low multiplier, whereas a strong prior belief enhances that impact via a factor greater than 1. As a result, NAG tends to assume that users will require more and better arguments to be persuaded to change their strongly held beliefs than should be necessary normatively.

Due to **overconfidence** people tend to exaggerate the probability of very likely events and the improbability of very unlikely events (Lichtenstein *et al.*, 1982). NAG uses this bias to select prior probabilities for propositions in the user model when it has not been given explicit information about these probabilities, but only frequency information also known to the user. NAG applies an S-curve to convert frequencies into user probabilities (Figure 5). For example, a very low frequency will be mapped into an even smaller prior probability. NAG does not use this S-curve directly when the base rate fallacy applies, since in that case both errors are dealt with at once.

The base rate fallacy is the tendency people have to ignore objective prior probabilities based on frequency data available to them and to replace these probabilities with a uniform probability distribution. A striking example of this is the cab problem described in (Tversky and Kahneman, 1982).

> A cab was involved in a hit and run accident at night. Two cab companies, the

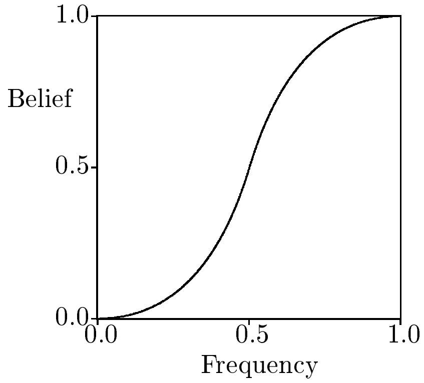

Figure 5: Overconfidence Curve

Green and the Blue, operate in the city. You are given the following data:

- 85% of the cabs in the city are Green and 15% are Blue.
- A witness identified the cab as Blue. The court tested the reliability of the witness under the same circumstances that existed on the night of the accident and concluded that the witness correctly identified each one of the two colors 80% of the time and failed 20% of the time.

By Bayes theorem the probability that the cab was Blue is 0.41 based on the evidence given. This is because the low prior probability of Blue Cabs (0.15) dominates the computation.[3] Nevertheless, most people presented with the story respond that the probability that the offending cab was blue is 0.8, a response that would be correct if the prior probabilities were uniformly distributed among the available options. This is characteristic of most people's response to many situations involving uncertainty — they flatten all prior probabilities. In order to accommodate this tendency, NAG computes implicit prior probabilities in the user model using a function that flattens the middle of the belief-frequency graph, thereby reflecting this bias towards uniform priors.

Using the odds-likelihood ratio method (Neapolitan, 1990) with the additional multiplicative factors as described above, we obtain the following update formula:

$$O(N_{new}|N_1, N_2, \ldots, N_n) = O(N_{new}) \times \prod_{i=1}^{n} \lambda_i \times \textit{belbias}_i \times \textit{overconf}_i \times \textit{baserate}_i,$$

where the three biasing factors for each inference are displayed separately, the evidence nodes N_i are conditionally independent given the "causal" node N_{new}, and the likelihood ratio is computed as $\lambda_i = P(N_i|N_{new})/P(N_i|\neg N_{new})$. The multiplicative factors for each of $\textit{belbias}_i$, $\textit{overconf}_i$ and $\textit{baserate}_i$ are either 1.0, if the particular cognitive weakness is not active for this inference, or derived as above if the weakness is active.[4] The propagation rule for uncertain evidence and the more general formulae used for propagation in singly connected causal networks given in (Neapolitan, 1990) are similarly modified by means of the three multiplicative factors.

4.1.1 Analyzing the Example

The argument that can be built at this stage has two main branches: from nodes N_1–N_3 to node N_{13} via N_{11} (Figure 3); and from node N_9 to N_{13} via $N_9 \rightarrow N_{12} \rightarrow N_{13}$ (Figure 4). According to the user model, the user is believed to attach great strength to the inferences $N_1 \rightarrow N_5 \rightarrow N_8$ and $N_3 \rightarrow N_6 \rightarrow N_8$, so propagation in the user model leads to an anticipated final belief in the goal node within the target range. However, the anticipated final belief in the goal node in the normative model falls short of the desired normative range.

NAG continues through Steps 7–9 of the Generation-Analysis algorithm, putting the newly added nodes connected to the goal node in the Argument Graph into the current context. On the next cycle through the algorithm (Step 2) a new round of semantic priming is performed. However, in the current example this does not lead to any further nodes being connected to the argument goal. As a consequence, during Step 3 only one new subgoal is tagged, node N_9, which was connected to the goal during the last cycle of the algorithm. When node N_9 is then passed to the Reasoning Agents in Step 4, the new link $N_{10} \rightarrow N_9$ is found and added to the Argument Graph. The resulting Argument Graph is returned to the Analyzer again (Step 5), which determines that the anticipated belief in the goal is now within the target ranges in both models.

The subgraph corresponding to a complete (non-enthymematic) argument generated for this example appears in Figure 6. Node N_2 is omitted because of its weak contribution to the goal, as explained in following section. The focusing and subgoal setting parts of the system managed to find an

[3] The computation being $(.8 \times .15)/(.8 \times .15 + .2 \times .85) = 0.41$.

[4] At present, the activation of a cognitive weakness or lack thereof is a tag that accompanies each inference in the user model BN. This will support the implementation of a user model which determines the activation or deactivation of a cognitive weakness depending on factors such as a user's expertise in a domain or his/her level of alertness.

acceptable solution to this example problem without even having to fully investigate every proposition in the Argument Graph (nodes N_4, N_7 and N_{10} were never passed to the Reasoning Agents), let alone every proposition in the complete BN in the user model and the complete BN in the normative model. Partial propagation over just the subnetworks corresponding to the Argument Graph in each model also ensured that inferences like [presence of igneous rocks] $\xrightarrow{evidence}$ [history of high temperature and pressure conditions], which turned out to be irrelevant for this argument goal, were not propagated over.

5 Presentation

After a successful Argument Graph has been built, so that the current Argument Graph now gives an anticipated belief in the goal proposition within the desired ranges in both the normative and user models, then comes the problem of how to present the argument to the user. NAG attempts to minimize the size of the current Argument Graph by searching it for the subargument with the fewest nodes which still yields a degree of belief in the goal proposition that falls inside the target ranges in both the normative model and the user model. During this process, it tries to generate an enthymematic argument by iteratively deleting nodes and invoking the Analyzer to determine whether the belief in the goal proposition in the now smaller Argument Graph still suffices. Frequently this process will be able to simplify the Argument Graph, since upon completion of the focusing-generation-analysis cycles, some of the propositions in the Argument Graph may be supported more strongly than is necessary for the argument to work.

NAG was designed to be a system which generates arguments and receives user arguments. At present, the front-end of the system is not implemented. However, a graphical interface is currently being constructed which allows the user to build and receive arguments in an annotated network form. Methods of rendering the system's output in English are being investigated.

6 Conclusion

NAG uses a series of focusing-generation-analysis cycles to build two Bayesian networks (one in the normative model and another in the user model) that contain the information required to construct a nice argument. Partial propagation, performed over the subnetworks in focus (the current Argument Graph), is used to estimate the impact of the resultant argument. Modified Bayesian update rules model three human cognitive weaknesses.

Any argumentation system must have access to a great deal of domain specific data if it is to generate and analyze arguments well. NAG is no exception, and consequently setting up a good domain, one with sufficient depth and richness to test NAG well, is not trivial. By allowing NAG to use existing knowledge sources where possible, via small Reasoning Agents written to match the various knowledge source types, we have endeavoured to at least partially mitigate this problem.

NAG has been tested on five sample scenarios which generate BNs containing up to about 50 nodes. The use of spreading activation to simulate attention, and the simplifications NAG employs to reduce the time taken to extend and propagate beliefs through the Bayesian subnetworks, lead to a significant reduction in argument generation times compared to trials run with the same BNs not using these techniques. These speed-up methods seem to have little effect on the resulting arguments.

Larger BNs and KBs, which are currently being built, will enable us to test more conclusively the effects of our modifications on the speed of the generation process and the quality of the arguments produced. These richer scenarios will also allow us to better test the effects of our modeling of human cognitive weaknesses. We are currently planning a variety of tests to evaluate the performance of our system. The graphical interface currently under construction and an English generator will be used to test the effect of arguments generated by NAG on users' beliefs. In addition, the English output will be used to compare NAG's arguments with those generated by people and to test how the order of presentation of the points in an argument affects users' beliefs.

Acknowledgments

This work was supported in part by Australian Research Council grant A49531227.

References

J.R. Anderson. 1983. *The Architecture of Cognition.* Cambridge, Massachusetts: Harvard University Press.

B. Baars. 1987. *A Cognitive Theory of Consciousness.* Cambridge University Press.

E. Charniak and R.P. Goldman. 1993. A Bayesian model of plan recognition. *Artificial Intelligence* **64**(1):53–80.

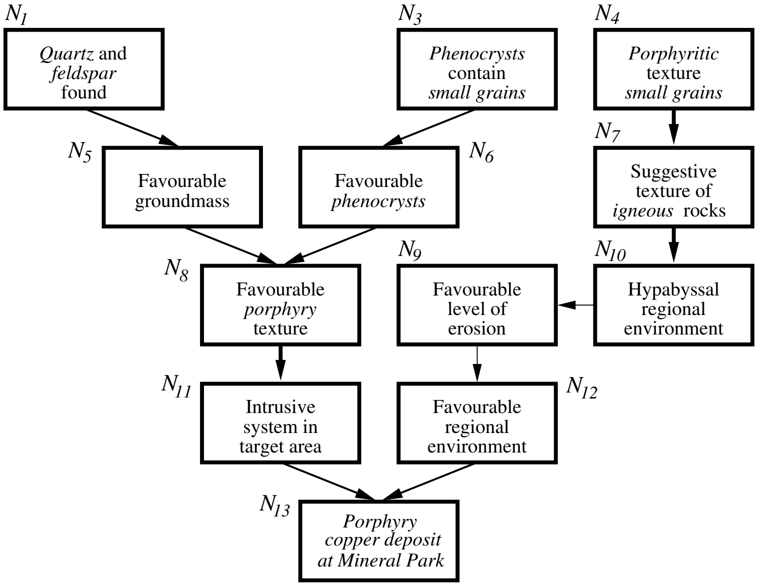

Figure 6: Final Sub-argument for the Porphyry Copper Deposit Example

G.F. Cooper. 1990. The Computational Complexity of Probabilistic Inference Using Belief Networks. *Artificial Intelligence* **42**:393–405.

R.O. Duda, P.E. Hart, P. Barrett, J.G. Gaschnig, K. Konolige, R. Reboh and J. Slocum. 1978. Development of the Prospector Consultation System for Mineral Exploration. Final Report for SRI Projects 5821 and 6415, SRI International.

J.St.B.T. Evans. 1989. *Bias in Human Reasoning: Causes and Consequences.* Hillsdale, NJ, USA.

J.R. Hobbs, M.E. Stickel, D.E. Appelt and P. Martin. 1993. Interpretation as abduction. *Artificial Intelligence* **63**(1–2):69–142.

H. Horacek. 1994. How to Avoid Explaining Obvious Things (Without Omitting Central Information). *Proceedings of the Eleventh European Conference on Artificial Intelligence*, 520–524, John Wiley & Sons.

D. Kahneman, P. Slovic and A. Tversky, editors. 1982. *Judgment under Uncertainty: Heuristics and Biases.* Cambridge University: Cambridge University Press.

K.B. Korb, R.S. McConachy and I. Zukerman. 1997. A Cognitive Model of Argumentation. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pages 400–405, Stanford University. Lawrence Erlbaum Associates.

Z. Li and B. D'Ambrosio. 1994. Efficient Inference in Bayes Nets as a Combinatorial Optimization Problem. *International Journal of Approximate Reasoning* **11**(1):55–81.

S. Lichtenstein, B. Fischhoff and L.D. Phillips. 1982. Calibrations of probabilities: The state of the art to 1980. In Kahneman *et al.*, pages 306-334.

S. Mehl. 1994. Forward Inferences in Text Generation. *Proceedings of the Eleventh European Conference on Artificial Intelligence*, 525–529, John Wiley & Sons.

R.E. Neapolitan. 1990. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms.* New York: John Wiley & Sons.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers.

R.H. Thomason, J.R. Hobbs and J.D. Moore. 1996. Communicative Goals. In *Proceedings of ECAI-96 Workshop – Gaps and Bridges: New Directions in Planning and NLG*, pages 7–12.

A. Tversky and D. Kahneman. 1982. Judgments of and by representativeness. In Kahneman *et al.*, pages 84-98.

M.A. Walker. 1996. The Effect of Resource Limits and Task Complexity on Collaborative Planning in Dialogue. *Artificial Intelligence* **85**(1-2):181–243.