Incorporating a User Model into an Information Theoretic Framework for Argument Interpretation[⋆]

Ingrid Zukerman, Sarah George and Mark George

School of Computer Science and Software Engineering
Monash University
Clayton, VICTORIA 3800, AUSTRALIA
{ingrid,sarahg}@csse.monash.edu.au, mark_thingy@yahoo.com

Abstract. We describe an argument-interpretation mechanism based on the Minimum Message Length Principle [1], and investigate the incorporation of a model of the user's beliefs into this mechanism. Our system receives as input an argument entered through a web interface, and produces an interpretation in terms of its underlying knowledge representation – a Bayesian network. This interpretation may differ from the user's argument in its structure and in its beliefs in the argument propositions. The results of our evaluation are encouraging, with the system generally producing plausible interpretations of users' arguments.

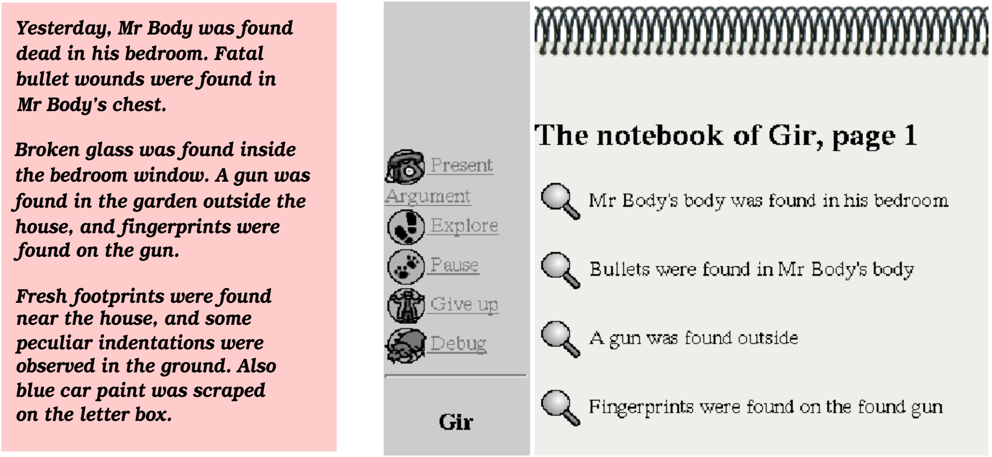
1 Introduction

Dialogue systems developed to date typically restrict users to a limited range of dialogue contributions. While this may be suitable for look-up systems, in other types of applications, e.g., tutoring systems, users would benefit from being able to present more complex responses. The discourse interpretation mechanism presented in this paper constitutes a significant step towards achieving this objective. Our mechanism interprets structured arguments presented by users in the context of an argumentation system.

This research builds on our previous work on BIAS – a *Bayesian Interactive Argumentation System* which uses Bayesian networks (BNs) [2] as its knowledge representation and reasoning formalism. In previous research, BIAS interpreted single-proposition rejoinders presented by a user after reading a system's argument [3]. Here BIAS interprets user arguments of arbitrary complexity, which may differ from BIAS' beliefs and inference patterns. The basic discourse-interpretation mechanism relies on the Minimum Message Length (MML) Principle [1] to evaluate candidate discourse interpretations [4]. The main contribution of this paper is in its principled incorporation of a user model into the discourse-interpretation mechanism.

In the following section, we describe our experimental set up. Next, we outline our knowledge representation formalism, and discuss the argument interpretation process. In Section 4, we provide an overview of our Minimum Message Length approach to discourse interpretation, and describe how user modeling information is incorporated into this formalism. The results of our evaluation are reported in Section 5. We then discuss related research, followed by concluding remarks.

[⋆] This research was supported in part by Australian Research Council grant A49927212.



Yesterday, Mr Body was found dead in his bedroom. Fatal bullet wounds were found in Mr Body's chest.

Broken glass was found inside the bedroom window. A gun was found in the garden outside the house, and fingerprints were found on the gun.

Fresh footprints were found near the house, and some peculiar indentations were observed in the ground. Also blue car paint was scraped on the letter box.

The notebook of Gir, page 1

- Mr Body's body was found in his bedroom
- Bullets were found in Mr Body's body
- A gun was found outside
- Fingerprints were found on the found gun

(a) Police report (b) Detective Gir's Notebook

Fig. 1. Police report and excerpt from Notebook

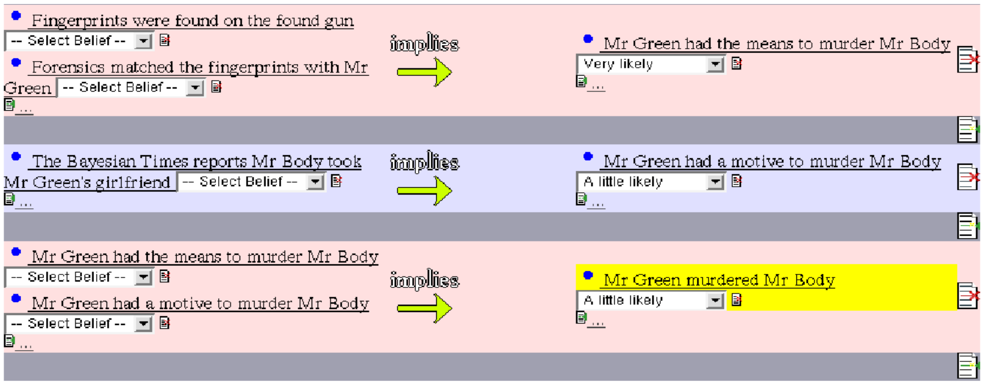
2 Experimental Set Up

Our experimental set up is similar to that of the system described in [3]. The user and the system are partners in solving a murder mystery, and obtain information by investigating the murder. However, in our current set up the user is a junior detective and the system is a desk-bound boss, who knows only what the user tells him. Thus, the user does all the leg-work, navigating through a virtual crime scene, making observations and interviewing witnesses, and reports periodically to the boss. These reports consist of successively evolving arguments for the main suspect's guilt or innocence.

The interaction with BIAS starts with the presentation of a police report that describes the preliminaries of the case for a particular scenario. The user then optionally explores the virtual scenario, recording in his/her *Notebook* information s/he finds interesting (this Notebook is employed by BIAS to build the user model, Section 4.2). Figure 1(a) shows the police report presented for the scenario used in this paper, and Figure 1(b) shows an excerpt of a user's Notebook after reading the report.

Upon completion of his/her investigation, the user builds an argument composed of a sequence of implications leading to the argument goal [Mr Green Killed Mr Body]. Each implication is composed of one or more antecedents and consequents, which, in the current implementation, are obtained by copying propositions from a drop-down menu or from the user's Notebook into slots in the argument-construction interface.[1] Figure 2 shows a screen-shot of the argument-construction interface, and an argument built by a particular user after she has read the police report, seen the newspaper and spoken to the forensic experts. Figure 3 shows the interpretation generated by BIAS for the argument in Figure 2. In it the system fills in propositions and relations where the user has made inferential leaps, and points out its beliefs and the user's (the user's input has been highlighted for clarity of presentation).

[1] An alternative version of our system accepts Natural Language (NL) input for antecedents and consequents. However, in order to isolate the contribution of the user modeling component, we have removed the NL capability from our current version.



- Fingerprints were found on the found gun
  -- Select Belief --
- Forensics matched the fingerprints with Mr Green -- Select Belief --
  ...

implies

- Mr Green had the means to murder Mr Body
  Very likely
  ...

- The Bayesian Times reports Mr Body took Mr Green's girlfriend -- Select Belief --
  ...

implies

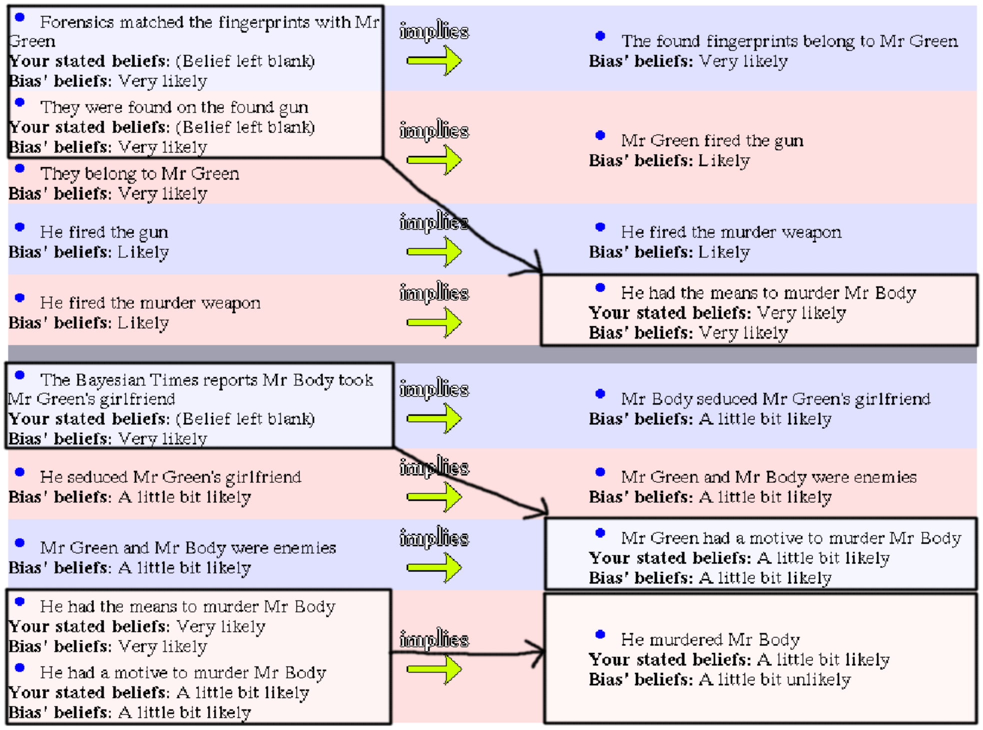
- Mr Green had a motive to murder Mr Body
  A little likely
  ...

- Mr Green had the means to murder Mr Body
  -- Select Belief --
- Mr Green had a motive to murder Mr Body
  -- Select Belief --
  ...

implies

- Mr Green murdered Mr Body
  A little likely
  ...

Fig. 2. Argument-construction screen and user's argument



- Forensics matched the fingerprints with Mr Green
  Your stated beliefs: (Belief left blank)
  Bias' beliefs: Very likely
- They were found on the found gun
  Your stated beliefs: (Belief left blank)
  Bias' beliefs: Very likely

implies

- The found fingerprints belong to Mr Green
  Bias' beliefs: Very likely

- They belong to Mr Green
  Bias' beliefs: Very likely

implies

- Mr Green fired the gun
  Bias' beliefs: Likely

- He fired the gun
  Bias' beliefs: Likely

implies

- He fired the murder weapon
  Bias' beliefs: Likely

- He fired the murder weapon
  Bias' beliefs: Likely

implies

- He had the means to murder Mr Body
  Your stated beliefs: Very likely
  Bias' beliefs: Very likely

- The Bayesian Times reports Mr Body took Mr Green's girlfriend
  Your stated beliefs: (Belief left blank)
  Bias' beliefs: Very likely

implies

- Mr Body seduced Mr Green's girlfriend
  Bias' beliefs: A little bit likely

- He seduced Mr Green's girlfriend
  Bias' beliefs: A little bit likely

implies

- Mr Green and Mr Body were enemies
  Bias' beliefs: A little bit likely

- Mr Green and Mr Body were enemies
  Bias' beliefs: A little bit likely

implies

- Mr Green had a motive to murder Mr Body
  Your stated beliefs: A little bit likely
  Bias' beliefs: A little bit likely

- He had the means to murder Mr Body
  Your stated beliefs: Very likely
  Bias' beliefs: Very likely
- He had a motive to murder Mr Body
  Your stated beliefs: A little bit likely
  Bias' beliefs: A little bit likely

implies

- He murdered Mr Body
  Your stated beliefs: A little bit likely
  Bias' beliefs: A little bit unlikely

Fig. 3. BIAS' interpretation of the user's argument

In order to evaluate the discourse interpretation capabilities of the system, in this paper we restrict users' interaction with the system to a single round. That is, a user reads the police report, optionally explores the virtual scenario, and generates an argument for Mr Green's guilt or innocence. BIAS then interprets the argument, and presents its interpretation back to the user for validation. The results of this validation are discussed in Section 5. In the future, the boss will present counter-arguments, point out flaws in the user's argument or make suggestions regarding further investigations.

3 Proposing Interpretations

The domain propositions and the relationships between them are represented by means of a BN. Each BN in the system can support a variety of scenarios, depending on the instantiation of the evidence nodes. For this paper we used an 85-node BN which represents a murder mystery (this BN is similar to that used in [3]).

BIAS generates an interpretation of a user's argument in terms of its own beliefs and inferences, which may differ from those in the user's argument. This may require adding propositions and relations to the argument structure proposed by the user, deleting relations from the user's argument, or postulating degrees of belief in the propositions in the user's argument which differ from those stated by the user. The procedure for proposing interpretations is described in [4]. Here we provide a brief overview, focusing on the structure of the interpretations (the beliefs in the propositions are obtained by performing Bayesian propagation).

Our system generates candidate interpretations for an argument by finding different ways of connecting the propositions in the argument – each variant being a candidate interpretation. This is done by (1) connecting the nodes in the argument, (2) removing superfluous nodes, and (3) building connected sub-graphs of the resultant graph.

Connecting nodes. This is done by retrieving from the domain BN neighbouring nodes to the nodes mentioned in the user's argument. Following [3], we perform two rounds of retrievals for each node in the user's argument. That is, we first retrieve its neighbours, and then its neighbours' neighbours from the domain BN. These retrieved neighbours are *inferred* nodes (they may have been previously added to the user model, or accessed now for the first time). We perform only two rounds of retrievals for each node in the user's argument in order to model small "inferential leaps", which the system would be expected to understand. As a result of this process, *mentioned* nodes that are separated by at most four inferred nodes in the domain BN will now be connected, but nodes that are further removed will remain unconnected. If upon completion of this process, a proposition in the user's argument is still unconnected, the system will have failed to find an interpretation (in the future, we will extend our MML-based formalism to consider interpretations that exclude one or more of the user's propositions).

Removing superfluous nodes. This is done by marginalizing out nodes that are not on a path between an evidence node and the goal node.

Building sub-graphs. BIAS derives all the interpretations for an argument by computing all the hyper-paths between two nodes (a hyper-path may comprise a single path or may be composed of more than one path between two nodes).

The Bayesian subnets generated in this manner are candidate interpretations of a user's argument in terms of BIAS' domain knowledge. However, these subnets alone do not always yield the beliefs stated by the user, as the user may have taken into account implicit assumptions that influence his/her beliefs. For instance, the argument in Figure 2 posits a belief of A Little Likely in Mr Green's guilt, while Bayesian propagation from the available evidence yields a belief of A Little **Un**likely. This discrepancy may be attributed to the user's lack of consideration of Mr Green's opportunity to murder Mr Body (her argument includes only means and motive), an erroneous assessment of Mr Green's opportunity, or an assessment of the impact of opportunity on guilt which differs from BIAS'. In the near future, our mechanism will consider the first two factors for neighbouring nodes of an interpretation (the third factor involves learning a user's Conditional Probability Tables – a task that is outside the scope of this project).

4 Using MML to Select an Argument Interpretation

The MML criterion implements Occam's Razor, which may be stated as follows: "If you have two theories which both explain the observed facts, then you should use the simplest until more evidence comes along".[2] MML distinguishes itself from other popular model-building approaches, such as Maximum Entropy, in that it provides a theoretical criterion for evaluating the goodness of a model, while the other approaches can be validated only empirically.

According to the MML criterion, we imagine sending to a receiver the shortest possible message that describes an NL argument. A message that encodes an NL argument in terms of an interpretation is composed of two parts: (1) instructions for building the interpretation from domain knowledge, and (2) instructions for rebuilding the original argument from this interpretation. These parts balance the need for a concise interpretation (Part 1) with the need for an interpretation that matches closely the original argument (Part 2). For instance, a concise interpretation yields a message with a short first part, but if this interpretation does not match well the original argument, the second part will be long. In contrast, a more complex interpretation which better matches the original argument may yield a shorter message overall. In any event, for an interpretation to be plausible, the message that encodes an NL argument in terms of this interpretation must be shorter than the message that transmits the words of the argument directly (if no such interpretation can be found, the argument is not being understood by the system).

The expectation from using the MML criterion is that in finding an interpretation that yields the shortest message for an NL argument, we will have produced a plausible interpretation, which hopefully is the intended interpretation. This interpretation is determined by comparing the message length of the candidate interpretations, which are obtained as described in Section 3.

In this section, we first review the MML encoding of an NL argument (a detailed description of this encoding appears in [4]). We then discuss the incorporation of a user model into this formalism.

4.1 MML Encoding

The MML criterion is derived from Bayes Theorem: $\Pr(D \& H) = \Pr(H) \times \Pr(D|H)$, where D is the data and H is a hypothesis which explains the data. An optimal code for an event E with probability $\Pr(E)$ has message length $\mathrm{ML}(E) = -\log_2 \Pr(E)$ (measured in bits). Hence, the message length for the data and a hypothesis is:

$$\mathrm{ML}(D \& H) = \mathrm{ML}(H) + \mathrm{ML}(D|H)$$

The hypothesis for which $\mathrm{ML}(D \& H)$ is minimal is considered the best hypothesis.

In our context, the data is the argument, and the hypothesis is the interpretation. Let *Arg* be a graph representing an argument (with antecedents pointing to consequents), and *SysInt* an interpretation generated by our system. Thus, we are looking for the *SysInt* which yields the shortest message length for

$$\mathrm{ML}(\mathit{Arg} \& \mathit{SysInt}) = \mathrm{ML}(\mathit{SysInt}) + \mathrm{ML}(\mathit{Arg}|\mathit{SysInt})$$

[2] The similarity between MML and Kolmogorov complexity, which is also an implementation of Occam's Razor, is discussed in [5].

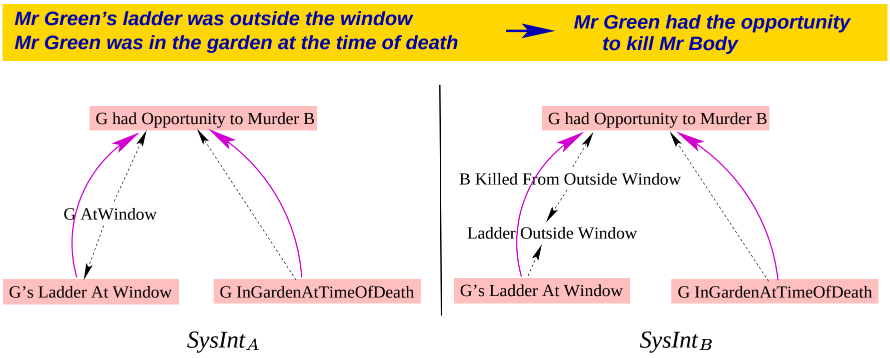

Mr Green's ladder was outside the window
Mr Green was in the garden at the time of death
Mr Green had the opportunity to kill Mr Body
G had Opportunity to Murder B
G AtWindow
G's Ladder At Window
G InGardenAtTimeOfDeath
$SysInt_A$
G had Opportunity to Murder B
B Killed From Outside Window
Ladder Outside Window
G's Ladder At Window
G InGardenAtTimeOfDeath
$SysInt_B$

Fig. 4. Interpretation of a simple argument

The first part of the message describes the interpretation, and the second part describes how to reconstruct the argument from the interpretation. Figure 4 illustrates the interpretation of a simple argument composed of two antecedents and one consequent. The Figure shows two *SysInt*s, with *Arg* superimposed on them (the nodes in *Arg* are shaded, and the links are curved). Since domain propositions (rather than NL sentences) are used to construct an argument, *Arg* can be directly obtained from the input.[3] *SysInt* is then derived from *Arg* by using the links and nodes in the domain BN to connect the propositions in the argument (Section 3). When the underlying representation has several ways of connecting between the nodes in *Arg*, then more than one candidate *SysInt* is generated, where each candidate has at least one *inferred* node that does not appear in the other candidates. For instance, the inferred nodes for the two interpretations in Figure 4 are [G At Window] for $SysInt_A$, and [Ladder Outside Window] and [B Killed From Outside Window] for $SysInt_B$; the inferred links are drawn with dashed lines.

After candidate interpretations have been postulated, the MML criterion is applied to select the best interpretation, i.e., the interpretation with the shortest message. The calculation of the message length takes into account (1) the size of an interpretation, and (2) the structural and belief similarity between the interpretation and the argument. These factors influence the components of the message length as follows.

- $\mathrm{ML}(SysInt)$ represents the probability of *SysInt*. According to the MML principle, concise interpretations (in terms of number of nodes and links) are more probable than more verbose interpretations.
- $\mathrm{ML}(Arg|SysInt)$ represents the probability that a user uttered *Arg* when s/he intended *SysInt*. According to this component, interpretations that are more similar to *Arg* are more probable than interpretations that are less similar to *Arg*. This probability depends on the structural similarity between *SysInt* and *Arg* (which in turn depends on the operations that need to be performed to transform *SysInt* into *Arg*), and on the closeness between the beliefs in the nodes in *Arg* and the beliefs in the corresponding nodes in *SysInt* (these beliefs are obtained by performing Bayesian propagation through *SysInt*; thus, different *SysInt*s may yield different beliefs in the consequents of an argument).

[3] This is not the case in the version of the system which takes NL input, since there may be more than one proposition that constitutes a reasonable interpretation for a sentence in an argument.

Table 1. Message length comparison of two interpretations

ML of	Factor	$\boldsymbol{SysInt_{\mathbf{A}}}$	$\boldsymbol{SysInt_{\mathbf{B}}}$	Shortest ML
SysInt	Size	4 nodes, 3 links	5 nodes, 4 links	$SysInt_A$
$Arg\|SysInt$	Structural similarity	more similar	less similar	$SysInt_A$
	Belief similarity	farther	closer	$SysInt_B$

Table 1 summarizes the effect of these factors on the message length for $SysInt_A$ and $SysInt_B$. $SysInt_A$ is simpler than $SysInt_B$, thus yielding a shorter message length for the first component of the message. $SysInt_A$ is structurally more similar to *Arg* than $SysInt_B$: $SysInt_A$ has 1 node and 2 links that are not in *Arg*, while $SysInt_B$ has 2 nodes and 3 links that are not in *Arg*. As a result, the structural aspect of $\mathrm{ML}(Arg|SysInt_A)$ has a shorter message length. In this example, we assume that the belief in [G had Opportunity to Kill B] in $SysInt_B$ is stronger than that in $SysInt_A$, and hence closer to the asserted consequent of the argument. This yields a shorter message length for the belief component of $\mathrm{ML}(Arg|SysInt_B)$. However, this is insufficient to overcome the shorter message length of $SysInt_A$ due to structural similarity and conciseness. Thus, although both interpretations of the user's argument are reasonable, $SysInt_A$ is preferred.

As can be seen from this account, the MML principle enables a discourse interpretation mechanism to weigh possibly conflicting considerations, and reach an understanding of the user's reasoning in terms of the system's domain knowledge.

4.2 Incorporating the User Model

The above formalism assumes that every proposition is equally likely to be included in an interpretation. However, this is not the case in reality. We postulate that interpretations comprising propositions familiar to the user (e.g., recently made observations, or propositions from his/her Notebook) are more probable than interpretations that include other domain propositions (although the user may still include unseen propositions of which s/he has thought independently).

In order to represent this observation in terms of the MML principle, the message that conveys *SysInt* must take into account the different probabilities associated with the domain propositions. These probabilities, which reflect the salience of propositions in the user's focus of attention, are modeled by means of two factors: (1) the type of access of a proposition, and (2) the frequency and recency of access. To derive these probabilities we need to provide numerical values for these factors.

Access type. Following [3], we distinguish between four types of access. Observations may be `seen` or `accepted`, and statements may be `mentioned` or `inferred`. `Seen` observations are those that the user has encountered but has not acknowledged, while `accepted` observations have been entered by the user in his/her Notebook. `Mentioned` statements have been explicitly included in the user's argument, while `inferred` statements were not mentioned by the user, but are incorporated by BIAS into an interpretation in order to connect the `mentioned` nodes (Section 3). We assign the following numerical strengths to our access categories.

$$Str(Node) = \begin{cases} A & \text{if } \texttt{accepted} \\ A & \text{if } \texttt{mentioned} \\ \max\{\frac{A}{F_S}, \frac{A}{\text{\#\_of\_props\_seen}+1}\} & \text{if } \texttt{seen} \\ \frac{A}{F_I} & \text{if } \texttt{inferred} \end{cases} \qquad (1)$$

where A, F_S and F_I are constants obtained by testing the system. According to this formula, the strength of a `seen` proposition is inversely proportional to the number of propositions viewed concurrently (e.g., read in the same page), and is always less than the strength of `accepted` propositions. The strength of `inferred` propositions is also low, because when BIAS includes an `inferred` node in an interpretation, it is uncertain that this node is intended by the user until s/he confirms the interpretation in question.

Frequency and recency. These factors are taken into account by means of the following function, which represents the level of activation of a node.

$$\sum_{i=1}^{n} [CurTime - TimeStmp_i + 1]^{-b} \qquad (2)$$

where n is the number of times a proposition was accessed, $b = 1$ is an empirically determined exponent, *CurTime* is the current time, and $TimeStmp_i$ is the time of the ith access. According to this formula, the level of activation of a node decays as a function of the time elapsed since its access. In addition, when a node is accessed, activation is added to the current accumulated (and decayed) activation. That is, there is a spike in the level of activation of the node, which starts decaying from that point again.

By combining these two factors we obtain the following formula for the score of a node (where Str_i is the strength of the ith access). This formula assigns a high score to nodes that were recently accepted or mentioned by a user.

$$Score(Node) = \sum_{i=1}^{n} Str_i(Node) \times [CurTime - TimeStmp_i + 1]^{-b} \qquad (3)$$

Probabilities for nodes. Equation 3 yields a score that reflects the salience of a node in the user's attentional focus. In order to derive a probability from this score, we normalize it as follows.

$$\Pr(Node_i) = \frac{Score(Node_i)}{\sum_{j=1}^{N} Score(Node_j)} \qquad (4)$$

where N is the number of nodes in the domain BN.

According to this formula, an `inferred` node that was not previously in the user model will have a low probability, which will incur a high message length. In contrast, an `inferred` node that was previously in the user model will have a higher score owing to previous accesses, and hence a higher probability.

To illustrate the effect of the user model on the argument interpretation process, let us reconsider the sample argument in Figure 4, and let us assume that [G At Window] is not in the user model, while [Ladder Outside Window] and [B Killed From Outside Window] are in the user model. In this case, a high score for these two propositions (obtained by accepting them recently or seeing them repeatedly) may overcome the factors in favour of *SysInt*$_A$, thereby making *SysInt*$_B$ the preferred interpretation.

5 Evaluation

The previous version of the system was evaluated by making it generate synthetic arguments, and then produce interpretations of its own arguments [4]. The results of this evaluation were encouraging, with the system generating plausible interpretations of its own arguments in 75% of the 5400 tried cases.

In this paper, we report the results of a formative evaluation with a few real users (10 computer-literate staff and students from Monash University). Our evaluation was conducted as follows. We introduced the users to our system, and explained its aims. We then encouraged them to explore the scenario, and when they were ready, they built an argument using the interface shown in Figure 2. BIAS then generated an interpretation of the argument, presenting it as shown in Figure 3. The users were asked to assess BIAS' interpretation under two conditions: before and after seeing a diagram of our 85-node BN. In the initial assessment, the users were asked to give BIAS' interpretation a score between 1 (Very UNreasonable) and 5 (Very Reasonable), and to optionally provide further comments. In the second assessment, the users were asked to re-assess BIAS' interpretation in light of the domain knowledge represented in the diagram. They were also asked to trace their preferred interpretation on the diagram (on paper).

Our users found the system somewhat daunting, and indicated that the interface for entering an argument was inconvenient. We believe that this was partly due to their lack of familiarity with the available domain propositions. That is, the users were faced with 85 new propositions, which they had to scan in order to determine whether they could use these propositions to express what they had in mind. Nonetheless, the users managed to construct arguments, which ranged in size from 2 propositions to 26, and gave a generally favourable assessment of BIAS' interpretations. Overall the average score of BIAS' interpretations was 4 before seeing the BN diagram and 4.25 after seeing the diagram. This indicates that a user's understanding of the system's domain knowledge may influence his/her interaction with the system, as the domain knowledge enables a user to better understand why a particular interpretation makes sense to the system.

The main lessons learned from this preliminary evaluation pertain to two aspects: (1) the interface, and (2) the use of BNs for discourse understanding. In order to improve the usability of the interface, we will integrate it with BIAS' NL module. It is envisaged that a solution combining menus and NL input will yield the best results. Our evaluation also corroborates the insights from Section 3 regarding the difficulties of taking into account users' assumptions during the argument interpretation process. However, the results of our evaluation are encouraging with respect to the use of the MML principle for the selection of interpretations, and the consultation of a user model during the selection process. In the future, we propose to conduct a comparative evaluation with and without the user model to determine its impact more accurately.

6 Related Research

As indicated in Section 1, our research builds on work described in [3, 4]. In this paper, we apply a principled approach based on the MML criterion [1] to select an interpretation for unrestricted arguments, instead of the heuristics used in [3] to select an interpretation for single-proposition rejoinders. We extend the work described in [4] in that we seamlessly incorporate a user model into the MML-based interpretation formalism.

The MML principle is a model-selection technique which applies information-theoretic criteria to trade data fit against model complexity. MML has been used in a variety of applications, several of which are listed in `http://www.csse.monash.edu.au/∼dld/Snob.application.papers`. In this paper, we demonstrate the applicability of MML to a high-level NL task.

BNs have been used in several systems that perform plan recognition, e.g., [6–8]. Charniak and Goldman's system [6] handled complex narratives, using a BN and marker passing for plan recognition. It automatically built and incrementally extended a BN from propositions read in a story, so that the BN represented hypotheses that became plausible as the story unfolded. Marker passing was used to restrict the nodes included in the BN. In contrast, we use domain knowledge to constrain our understanding of the propositions in a user's argument, and apply the MML principle to select a plausible interpretation. Gertner *et al.* [7] used a BN to represent the solution of a physics problem. After observing an action performed by a student, their system postulated candidate interpretations (like BIAS' *SysInt*), each comprising subsequent actions. In contrast, instead of being given one action at a time, BIAS is presented with a complete argument. Hence, it must also consider the fit between all the argument propositions and the interpretation ($Arg|SysInt$). Finally, Horvitz and Paek's system [8] handled short dialogue contributions, and used BNs at different levels of an abstraction hierarchy to infer a user's goal in information-seeking interactions with a Bayesian Receptionist. In addition, they employed decision-theoretic strategies to guide the progress of the dialogue. We expect to use such strategies when our system engages in a full dialogue with users.

7 Conclusion

We have offered a mechanism based on the MML principle that generates interpretations of extended arguments in the context of a BN. The MML principle provides a theoretically sound framework for selecting a plausible interpretation among candidate options. This framework enables us to represent structural discrepancies between the underlying, detailed domain representation and the more sparse arguments produced by people (which typically contain inferential leaps). The user modeling information incorporated into the MML framework allows the interpretation mechanism to take into account the manner of acquisition of domain propositions, and their frequency and recency of access. The results of our formative evaluation are encouraging, supporting the application of the MML principle for argument interpretation.

References

1. Wallace, C., Boulton, D.: An information measure for classification. The Computer Journal **11** (1968) 185–194
2. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann Publishers, San Mateo, California (1988)
3. Zukerman, I.: An integrated approach for generating arguments and rebuttals and understanding rejoinders. In: UM01 – Proceedings of the Eighth International Conference on User Modeling, Sonthofen, Germany (2001) 84–94
4. Zukerman, I., George, S.: Towards a noise-tolerant, representation-independent mechanism for argument interpretation. In: COLING 2002 Proceedings – the 19th International Conference on Computational Linguistics, Taipei, Taiwan (2002) 1170–1176
5. Wallace, C., Dowe, D.: Minimum message length and Kolmogorov complexity. The Computer Journal **42** (1999) 270–283
6. Charniak, E., Goldman, R.P.: A Bayesian model of plan recognition. Artificial Intelligence **64** (1993) 50–56
7. Gertner, A., Conati, C., VanLehn, K.: Procedural help in Andes: Generating hints using a Bayesian network student model. In: AAAI98 – Proceedings of the Fifteenth National Conference on Artificial Intelligence, Madison, Wisconsin (1998) 106–111
8. Horvitz, E., Paek, T.: A computational architecture for conversation. In: UM99 – Proceedings of the Seventh International Conference on User Modeling, Banff, Canada (1999) 201–210