Modeling Suppositions in Users' Arguments*

Sarah George and Ingrid Zukerman and Michael Niemann

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

Abstract. During conversation, people often make assumptions or suppositions that are not explicitly stated. Failure to identify these suppositions may lead to mis-communication. In this paper, we describe a procedure that postulates such suppositions in the context of the discourse interpretation mechanism of BIAS – a *Bayesian Interactive Argumentation System*. When a belief mentioned in a user's discourse differs from that obtained in BIAS' user model, our procedure searches for suppositions that explain this belief, preferring suppositions that depart minimally from the beliefs in the user model. Once a set of suppositions has been selected, it can be presented to the user for validation. Our procedure was evaluated by means of a web-based trial. Our results show that the assumptions posited by BIAS are considered sensible by our trial subjects.

1 Introduction

During conversation, people often make assumptions or suppositions that are not explicitly stated. The identification of these suppositions is important in order to understand the intentions or reasoning of one's conversational partner, and to provide cooperative responses. For instance, if someone says "Jack is tall, so Jill must be tall", s/he is probably assuming that Jack and Jill are related. This assumption must be taken into account in order to respond cooperatively. In this example, rather than responding "I disagree, Jill may or may not be tall", it would be more helpful to say "Actually, Jack and Jill are not related, so we can't infer that Jill is tall".

In this paper, we describe a procedure that postulates such suppositions in the context of the discourse interpretation mechanism of BIAS – a *Bayesian Interactive Argumentation System* [8, 7]. This mechanism receives as input arguments for a goal proposition, and generates *interpretations*.

An interpretation is a representation of what an interlocutor said in terms of the mental model maintained by the addressee. When the addressee is a computer, this representation is constrained by the knowledge representation and reasoning formalism employed by the system, and by the purpose for which the system is used. The interpretations generated by the version of BIAS described in [8, 7] consist of propositions associated with degrees of belief, and relations between propositions. For example, if a user said "If I walk to the main road, then I'll probably be in Sydney tomorrow", one possible interpretation would be "*WalkMainRoad* → ***TakeBus*** → *ArriveSydney* [Likely]", and another would be "*WalkMainRoad* → ***HitchRide*** → *ArriveSydney* [Likely]".

* This research was supported in part by the ARC Centre for Perceptive and Intelligent Machines in Complex Environments. The authors thank David Albrecht and Yuval Marom for their help with the analysis of the evaluation results.

The procedure described in this paper incorporates suppositions into such interpretations in order to account for the beliefs stated by a user. For example, if the user had been previously discussing the perils of hitchhiking, and then said "If I walk to the main road, I'll have to hitch a ride to Sydney", the system could posit that the user is supposing that no buses are available. If such a supposition departs significantly from the beliefs recorded in the user model, it is presented to the user for confirmation, whereupon it is incorporated into the user model.

In the next section, we discuss related research. Section 3 outlines our interpretation-generation process, and Section 4 describes our mechanism for positing suppositions. We then present a user-based evaluation of this mechanism, and concluding remarks.

2 Related Research

An important aspect of discourse understanding involves filling in information that is omitted by the interlocutor. In our previous work, we have considered *inferential leaps*, where BIAS filled in intermediate reasoning steps left out by a user [8, 7], and *unstated premises*, where BIAS postulated which premises from the user model were considered by the user, but omitted from his/her argument [9]. In this paper, we consider *suppositions*, which according to the Webster dictionary "consider as true or existing what is not proved". Suppositions are beliefs that *differ* from those in the user model, but are posited by the system to account for the beliefs expressed in the user's argument.

Several researchers have considered *presuppositions*, a type of suppositions implied by the wording of a statement or query [3, 5, 4, 2]. For instance, "How many people passed CS101?" presupposes that CS101 was offered and that students were enrolled in it [3]. Mercer [4] used default logic together with lexical information to identify a speaker's presuppositions. Gurney *et al.* [2] used active logic plus syntactic and lexical information to update the discourse context presupposed by an utterance. Kaplan [3] considered the information in a database and applied language-driven inferences to identify *presumptions* in database queries, and generate indirect cooperative responses, e.g., "CS101 was not offered" rather than "nobody passed CS101". Motro [5] extended this work using information about the database, such as integrity constraints, in addition to the information in the database.

The presuppositions considered by these researchers are typically few and can be unequivocally inferred from the wording of a single statement. In contrast, the suppositions considered in this paper are postulated to justify the relations between statements made by a user, and differ from what our system thinks that the user believes. Furthermore, there may be several alternative suppositions that explain a user's statements, and their probability depends on the other beliefs held by a user.

3 Outline of BIAS

BIAS uses Bayesian networks (BNs) [6] as its knowledge representation and reasoning formalism.[1] Our domain of implementation is a murder mystery, which is represented by a 32-node binary BN. That is, each node in the BN may be set to either True or False. In addition, an unobserved node may remain unset (with a probability between 0 and 1 inferred by means of Bayesian propagation).

[1] However, BNs are not essential. Our mechanism requires a set of propositions that represent the system's domain knowledge, and a representation of relations between propositions.

GreenInGardenAt11 [VeryLikely]
AND
MurderWeaponFiredByGreen [Likely]

IMPLIES

GreenMurderedBody [BetterThanEvenChance]

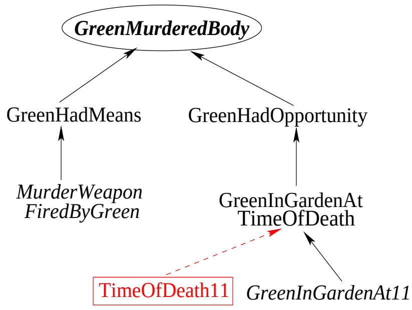

Fig. 1. Sample argument and interpretation

In the context of a BN, an interpretation consists of the tuple $\{SC, IG\}$, where SC is a *supposition configuration*, and IG is an *interpretation graph*.

– A **Supposition Configuration** is a set of suppositions made by BIAS to account for the beliefs in a user's argument.
– An **Interpretation Graph** is a subnet of the domain BN which links the nodes that correspond to the antecedents in an argument to the nodes that correspond to the consequents. Each node is associated with a degree of belief.

Figure 1 shows a sample argument (left-hand side) and interpretation (the Bayesian subnet on the right-hand-side). The argument is composed of propositions (obtained from a menu in an argument-construction interface [7]) linked by argumentation connectives. The italicized nodes in the Bayesian subnet are those mentioned in the argument, and the boxed node is a supposition (posited by the system) that accounts for the beliefs in the argument. If the time of death is unknown according to the user model, then GreenInGardenAt11 does not necessarily imply that Mr Green was in the garden at the time of death, yielding a belief of LessThanEvenChance in GreenMurderedBody. In order to account for the user's belief of BetterThanEvenChance for this consequent, BIAS posits that the user supposes TimeOfDeath11=True.

The problem of finding the best interpretation $\{SC, IG\}$ is exponential. Hence, we use the anytime algorithm in Figure 2 for this task [1]. This algorithm activates the following three modules until it runs out of time (after 20 seconds), retaining the top N (=4) interpretations at any point in time: one module for proposing a supposition configuration, one for proposing an interpretation graph, and one for evaluating the resultant interpretation.[2] This algorithm typically generates a few supposition configurations (Section 4), and several interpretation graphs for each supposition configuration.

An interpretation is evaluated by calculating its posterior probability, where the best interpretation is that with the highest posterior probability.

$$\mathit{SysIntBest} = \text{argmax}_{i=1,\ldots,n} \Pr(SC_i, IG_i | \mathit{UserArg})$$

where n is the number of interpretations.

[2] We have also implemented a module that matches Natural Language (NL) sentences in an argument with nodes in the domain BN, This module, which should be called before the other modules, is not part of the version of BIAS described here, where the propositions in an argument are copied from a menu.

Algorithm *GenerateInterpretations(UserArg)*
while {there is time}
{
1. Propose a supposition configuration SC that accounts for the beliefs stated in the argument.
2. Propose an interpretation graph IG that connects the nodes in *UserArg* under supposition configuration SC.
3. Evaluate interpretation $\{SC, IG\}$.
4. Retain top N (=4) interpretations.

}

Fig. 2. Anytime algorithm for generating interpretations

After applying Bayes rule and making independence assumptions, we obtain

$$\textit{SysIntBest} = \mathrm{argmax}_{i=1,\ldots,n}\{\Pr(\textit{UserArg}|SC_i, IG_i) \times \Pr(IG_i) \times \Pr(SC_i)\} \quad (1)$$

This calculation implements Occam's Razor, which may be stated as follows: "If you have two theories both of which explain the observed facts, then you should use the simplest until more evidence comes along". This principle balances data fit against model complexity. The data fit component ($\Pr(\textit{UserArg}|SC_i, IG_i)$) reflects how well an interpretation matches a user's argument both in structure and beliefs. The model complexity component reflects the simplicity of the interpretation, or how easily it can be derived from existing information. $\Pr(IG_i)$, the prior probability of an interpretation graph, reflects how easy it is to obtain this graph from the domain BN (e.g., small graphs are easy to derive); and $\Pr(SC_i)$, the prior probability of a supposition configuration, indicates how close are these suppositions to the beliefs in the user model. The calculation of $\Pr(\textit{UserArg}|SC_i, IG_i)$ is based on $\Pr(\textit{UserArg}|IG_i)$ (the calculation of $\Pr(\textit{UserArg}|IG_i)$ and $\Pr(IG_i)$ is described in [7], where we considered only interpretation graphs, not suppositions). In this paper, we describe the calculation of $\Pr(SC_i)$ and the influence of suppositions on the probability of an interpretation (Section 4).

4 Positing Suppositions

As stated in Section 3, the nodes in our BNs are binary. Hence, the possible supposition states are: SET TRUE – suppose that a node is True; SET FALSE – suppose that a node is False; and UNSET – suppose that a node has not been observed (i.e., ignore any evidence supplied by this node). Making a supposition may strengthen the influence of a node on its consequents, as shown in the example in Figure 1, or weaken it.

A supposition configuration describes the state of every node in the BN, hence there are 3^N such configurations (where N is the number of nodes in the BN). Since the number of nodes in the BNs implemented in BIAS ranges between 32 and 85, we cannot consider all possible supposition configurations, and we certainly cannot combine them with large numbers of interpretation graphs in the next step of algorithm *GenerateInterpretations*. We therefore find promising supposition configurations by generating only a limited number (=200) of supposition configurations that are close to the beliefs in the user model, and selecting from these the best three configurations as the basis for the generation of interpretation graphs. This is done by applying algorithm *GetSuppositionConfig* (Figure 3).

Algorithm *GetSuppositionConfig*, which is called in Step 1 of algorithm *GenerateInterpretations*, receives as input an argument *UserArg* and returns a supposition con-

Algorithm *GetSuppositionConfig(UserArg)*

1. If *SuppositionConfigList* is empty
   (a) Call *MakeNewConfig(Supposition)* K (=200) times, where each time *MakeNewConfig* returns the best supposition configuration.
   (b) Assign the top k (=3) supposition configurations to *SuppositionConfigList*.
2. Select an element from *SuppositionConfigList* at random.
3. Return the chosen configuration.

Algorithm *MakeNewConfig(ConfigType)*

1. If the priority queue is empty, ***propose an initial configuration***, ***calculate its probability***, and add the configuration and its probability to the priority queue.
2. Remove the first configuration from the queue.
3. ***Generate the children of this configuration***, ***calculate their probability***, and insert them in the queue so that the queue remains sorted in descending order of the probability obtained for a configuration.
4. Return the chosen (removed) configuration.

Fig. 3. Algorithm for generating suppositions

figuration randomly selected from a short-list of k (=3) configurations. This short-list, which is denoted *SuppositionConfigList*, is generated by calling *MakeNewConfig(Supposition)* K (=200) times, and selecting the best three configurations.

Algorithm *MakeNewConfig*, which is called in Step 1(a) of *GetSuppositionConfig*, maintains a priority queue of configurations and their probabilities. Each time it is called, it removes the configuration at the top of the queue (which has the highest probability), generates its "child configurations" (derived from the removed one), inserts them in the queue according to their probability, and returns the removed configuration.[3] The bold-italicized segments of the algorithm are explained later in this section.

We have adopted this process for the generation of supposition configurations, because observations of our system's behaviour indicate that there are only a few promising supposition configurations among the many possible options, but these configurations generally do not follow a monotonic pattern. Hence, a procedure that just descends a priority queue will not yield good results reliably. Further, trials performed during system development show that the top 200 supposition configurations (obtained by repeatedly accessing a priority queue) provide a suitable basis for selecting three promising configurations.

The generation of supposition configurations and their children employs a structure called *Supposition Score Table*, which maps nodes to suppositions (Table 1). Each column in the Supposition Score Table corresponds to a node in the BN. Each node is associated with a list of $<$supposition: probability$>$ pairs – one pair for each supposition – sorted in descending order of probability. Each pair represents the probability of making this supposition about the node in question, which is obtained by applying the following heuristics:

– *No change is best:* There is a strong bias towards not making suppositions.
– *Users are unlikely to change their mind about observed evidence:* If a user has observed a node (e.g., its value is True or False), s/he is unlikely to change his/her belief in this node.

[3] This algorithm is also used to generate interpretation graphs and node configurations that match NL sentences, but here we focus on its use for generating supposition configurations.

Table 1. Sample Supposition Score Table

$node_1$		$node_2$		...	$node_{32}$	
UNSET:	0.7	SET TRUE:	0.8	...	UNSET:	0.7
SET TRUE:	0.21	UNSET:	0.15	...	SET TRUE:	0.15
SET FALSE:	0.09	SET FALSE:	0.05	...	SET FALSE:	0.15

Table 2. Probability of making suppositions

	Node has been observed by the user		**Node has not been observed by the user**
Probability	**Node = FALSE**	**Node = TRUE**	
Pr(UNSET)	0.15	0.15	$\Pr_{unset}$ (=0.7)
Pr(SET FALSE)	0.8	0.05	$\Pr(\text{FALSE}) \times \Pr_{floating} + \Pr_{fixed}$
Pr(SET TRUE)	0.05	0.8	$\Pr(\text{TRUE}) \times \Pr_{floating} + \Pr_{fixed}$

– *Small changes in belief are better than large changes:* If a node that is left unset has a propagated value of 0.9, then it is more likely that the user is assuming it True than if the propagated value was 0.6.

These heuristics are implemented by means of the probabilities in Table 2. The left side of Table 2 specifies the probabilities of making suppositions about nodes that have been observed by the user. For example, if the user knows that GreenInGardenAt11=True, then the probability of setting this node to True (leaving it unchanged) is 0.8, the probability of unsetting this node is 0.15, and the probability of setting it to False is 0.05. The right side of Table 2 specifies the probabilities of making suppositions about nodes which have not been observed by a user (i.e., nodes that are unset). As per the above heuristics, the bulk of the probability mass is allocated to leaving a node unset. The remainder of the probability mass is allocated in proportion to the propagated probability of the node ($\Pr_{floating} = 0.2$ is used to normalize this component). However, we include a fixed component of $\Pr_{fixed} = 0.05$ to ensure that some probability mass is allocated to every value (i.e., the probability of setting a node to True or False can not go below 0.05). For instance, if the propagated belief of unobserved node GreenHadMeans is $\Pr(\text{GreenHadMeans}) = 0.8$, then the probability of leaving it unset is 0.7, the probability of setting it to True is $0.8 \times 0.2 + 0.05 = 0.21$ and the probability of setting it to False is $0.2 \times 0.2 + 0.05 = 0.09$.

The Supposition Score Table is used by elements of algorithm *MakeNewConfig* (Figure 3) to generate supposition configurations as follows.

Propose an initial configuration (Step 1 of MakeNewConfig). Select the first row from the Supposition Score Table. This yields supposition configuration $\{$*node*$_1$: UNSET, *node*$_2$: SET TRUE, ..., *node*$_{32}$: UNSET$\}$ for the Supposition Score Table in Table 1.

Generate the children of a configuration (Step 3). The ith child is generated by moving down one place in column i in the Supposition Score Table, while staying in the same place in the other columns. For the Supposition Score Table in Table 1, this yields $\{$<u>*node*$_1$: SET TRUE</u>, *node*$_2$: SET TRUE, ..., *node*$_{32}$: UNSET$\}$, $\{$*node*$_1$: UNSET, <u>*node*$_2$: UNSET</u>, ..., *node*$_{32}$: UNSET$\}$, ..., where the underlined node-supposition pair is the element being replaced in the parent supposition configuration.

Calculate the probability obtained for a configuration (Steps 1 and 3). According to Equation 1 (Section 3), the probability of an interpretation is given by

$$\Pr(\textit{UserArg}|SC_i, IG_i) \times \Pr(IG_i) \times \Pr(SC_i)$$

The probability of a supposition configuration, $\Pr(SC_i)$, is the product of the probabilities of the entries in the Supposition Score Table for the configuration in question. For instance, the initial configuration selected above has probability $0.7 \times 0.8 \times \ldots \times 0.7$, and configuration $\{\underline{\textit{node}_1\text{: SET TRUE}}$, *node*$_2$: SET TRUE, ..., *node*$_{32}$: UNSET$\}$ has probability $\underline{0.21} \times 0.8 \times \ldots \times 0.7$. Thus, the more SC_i departs from the beliefs in the user model, the lower is $\Pr(SC_i)$, thereby reducing the overall probability of the interpretation.

However, recall that $\Pr(\textit{UserArg}|SC_i, IG_i)$ depends both on the structural match between IG_i and *UserArg* and the match between the beliefs in IG_i (influenced by the suppositions in SC_i) and those in *UserArg*. Thus, if SC_i yields a better match between the beliefs in the interpretation and those in the user's argument, then the probability of Pr(beliefs in *UserArg*$|SC_i, IG_i)$ increases. As a result, the "cost" incurred by the suppositions in SC_i may be overcome by the "reward" resulting from the better match between the beliefs. This cost-reward balance is represented by the product Pr(beliefs in *UserArg*$|SC_i, IG_i) \times \Pr(SC_i)$, which determines the position of configuration SC_i in the priority queue maintained by algorithm *MakeNewConfig* (this product is also used to calculate Equation 1). Thus, the configurations that yield the best cost-reward balance *among those inspected until now* are at the top of the queue (children that are more promising may be discovered next time *MakeNewConfig* is called).

Our process for generating supposition configurations proposes promising configurations in terms of improvements in the belief match between an argument and an interpretation. However, it does not take into account other types of interactions which may cause locally optimal supposition configurations and interpretation graphs to combine into interpretations that are sub-optimal as a whole or even invalid. For example, if a user says $A \rightarrow C$ and the most direct path between A and C in the BN is $A \rightarrow B \rightarrow C$, then if B has been set to True in the user model, this path is *blocked* [6], as B prevents A from influencing C (which does not reflect the reasoning employed in the user's argument). Thus, the shortest interpretation graph together with the best supposition configuration (which retains the beliefs in the user model) yield an invalid interpretation. In this case, unsetting the value of B (supposing that it was not observed) makes the above interpretation valid. However, this may still not be the best interpretation, as there may be a longer interpretation, e.g., $A \rightarrow D \rightarrow E \rightarrow C$, which is not blocked and requires no suppositions. Such global effects are considered during the evaluation of an interpretation as a whole (Step 3 of algorithm *GenerateInterpretations*).

5 User Evaluation

Our evaluation of the module for postulating suppositions was conducted as follows. Using a Web interface, we presented four scenarios: Crimson and Lemon (Figure 4), Sienna and Mauve. These scenarios test various supposition alternatives as follows. The Crimson and Sienna scenarios required supposing that a node is True in order to strengthen the belief in the goal proposition of an argument; the Lemon scenario required a True supposition in order to unblock a path; and the Mauve scenario required unsetting or "forgetting" the value a node to weaken the belief in the goal proposition of an argument. Each scenario contained background evidence (not shown in Figure 4)

CRIMSON SCENARIO	LEMON SCENARIO
We think that *If forensics matched the bullets in Mr Body's body with the found gun, then the suspect Mr Green* ***possibly*** *had the means to murder Mr Body.*	**We think that** *If broken glass was found, then Mr Body's window* ***probably wasn't*** *broken from outside.*
If someone says *Forensics matching the bullets with the found gun means Mr Green* ***very probably*** *had the means to murder Mr Body.*	**If someone says** *Broken glass being found means Mr Body's window* ***probably was*** *broken from outside.*
Then it would be reasonable to think that they are assuming	
S1) Mr Green fired the gun found in the garden.	S1) Broken glass was found inside the window.
S2) The gun found in the garden is the murder weapon.	S2) The suspect Mr Green argued with Mr Body last night.
S3) Mr Green fired the murder weapon.	S3) Mr Body was killed from outside the window.
S4) Mr Green murdered Mr Body.	S4) Mr Green was in the garden at the time of death.
S5) None of the above are suitable as assumptions. A more likely assumption (in light of what our system understands) is .... [LINK TO LIST OF PROPOSITIONS]	
S6) It is not appropriate to think they are assuming anything.	

Fig. 4. Crimson and Lemon scenarios for user trials

and two versions of a short argument for a goal proposition in our BN. One version (denoted "We think that") stated the belief obtained for the goal proposition by performing Bayesian propagation from the evidence, and the other version (denoted "If someone says") gave a different belief for this proposition. The trial subjects were then asked to determine what this "someone" may be assuming in order to account for his/her belief in the goal proposition.

We have used this "indirect" evaluation method (instead of having subjects interact freely with the system), because we wanted to remove extraneous factors (such as interface usability) from the evaluation, and we wanted to focus on a particular behaviour of the system (the postulation of suppositions) that does not occur for every argument.

Since the purpose of our evaluation is to determine whether BIAS generates sensible suppositions in the context of its domain knowledge, we needed to limit the suppositions available to our trial subjects to the propositions known to BIAS. However, at the same time, we did not wish to burden our subjects with the need to look through BIAS' knowledge base to find out what BIAS knows. Additionally, we wanted to allow respondents some freedom to state their views, if they disagreed with BIAS' suppositions. These requirements were addressed by presenting our subjects with the following options (Figure 4): (S1-S4) a list of four candidate suppositions (one was the top supposition recommended by BIAS, and most of the others were considered by BIAS to be reasonable options); (S5) an option to include an alternative supposition (the subjects were provided a link to a list containing the propositions in the BN, but could also write a supposition of their own); and (S6) an option to state that they didn't believe that any suppositions were required.

The order of presentation of the suppositions was randomized across the scenarios. However, for the discussion in this paper, BIAS' preferred supposition is always S1. The trial subjects had to award a rank of 1 to one option and could optionally rank additional alternatives (with inferior ranks). This allowed respondents to ignore suppositions that

Table 3. Ranking of candidate suppositions for the four scenarios

LEMON SCENARIO	**Total**	**R1**	**R2**	**Other**	**MAUVE SCENARIO**	**Total**	**R1**	**R2**	**Other**
Supposition S1	30	30	0	0	Supposition S1	25	19	4	2
Supposition S3	11	0	11	0	Supposition S5	14	8	5	1
Total responses	51	34	12	5	Total responses	62	34	16	12
CRIMSON SCENARIO	**Total**	**R1**	**R2**	**Other**	**SIENNA SCENARIO**	**Total**	**R1**	**R2**	**Other**
Supposition S1	20	10	8	2	Supposition S1 + S3	30	20	6	4
Supposition S2	18	11	3	4	Supposition S4	16	7	6	3
Total responses	75	34	21	20	Total responses	69	34	19	16

didn't make sense to them, while enabling them to include more than one option that seemed reasonable. At the same time, the results obtained by this method enable us to determine whether BIAS' suppositions are considered sensible, even if they are not our subjects' top-ranked preferences.

Our four scenarios were considered by 34 participants. Many of the respondents had not been exposed to BIAS previously and were from outside the industry. The responses for the Lemon and Mauve scenarios were clear cut, while the responses for the Crimson and Sienna scenarios were more ambiguous, but still positive. The results for these scenarios are shown in Table 3. The top rows for each scenario contain the suppositions that were preferred by the trial subjects, and the bottom row lists the total responses for each scenario and for the different ranks (recall that the only rank that had to be given was 1). The columns contain the total number of respondents that ranked a supposition (Total), and the number of respondents that ranked it first (R1), second (R2) or gave it a lower rank (Other). Our results are summarized below.

- Supposition S1 was clearly the most favoured choice for the **Lemon** scenario, with 30 of the 34 respondents ranking it first. Supposition S3 was clearly the next best choice, with 11 trial subjects giving it a rank of 2.
- Supposition S1 was the preferred choice for the **Mauve** scenario, with 19 of the 34 respondents giving it a rank of 1. The next best choice was the Alternate Supposition, with only 8 subjects ranking it first. There were no clear preferences for rank 2, with all options receiving this rank at least once, but never more than five times.
- Suppositions S1 and S2 for the **Crimson** scenario were similarly ranked (each ranked first by about 1/3 of the subjects), with Supposition S1 being favoured slightly over S2, but not significantly so. The other options were ranked first only by a few trial subjects.
- The responses for the **Sienna** scenario presented us with a special case. The results of the first 22 responses and the comments provided by our trial subjects indicated that there was some confusion due to the wording of the instructions and the fact that, unlike the other scenarios, the Sienna scenario included a True and False version of the same node (Supposition S1 was "The time of death was 11 pm last night" and S3 was the negation of S1). Further, Supposition S3 supports the "We think that" version, while S1 supports the "If someone says" version. As a result, most of the respondents were divided between giving a rank of 1 to Supposition S1 or Supposition S3. Nonetheless, the main outcome from this scenario is that regardless of how the respondents read it, they clearly felt that a supposition had to be made about the "time of death" node, which was ranked first by 20 of the 34 respondents.

– Overall, very few trial subjects felt that no suppositions were warranted (9 for all the scenarios combined). Further, BIAS' preferred supposition was consistently ranked first or second, with its average rank being the lowest (best) among all the options.

These results justify the importance of making suppositions, and indicate that the suppositions made by BIAS not only are considered reasonable by people, but also have significant support.

6 Conclusion

We have offered a mechanism that postulates suppositions made by users in their arguments, and have shown how this mechanism is incorporated into our argument interpretation process. Our mechanism includes a procedure for generating suppositions, a method for calculating the probability of a set of suppositions, and a formalism for incorporating this probability into the probability of an interpretation.

An important feature of our system is its stability, in the sense that it does not match spurious beliefs (that don't follow a "sensible" line of reasoning). That is, the system will posit a supposition for a node only if it yields a payoff, i.e., a substantially better match between the beliefs in an interpretation and those in a user's argument. This behaviour is a result of BIAS' inherent reluctance to posit suppositions, combined with its reliance on a rigorous reasoning formalism, such as BNs, which requires the beliefs in the system to be consistent.

Finally, the results of our evaluation show that our trial subjects found BIAS' suppositions to be both necessary and reasonable, with its preferred suppositions being top-ranked or top-2 ranked by most subjects.

References

1. Sarah George and Ingrid Zukerman. An anytime algorithm for interpreting arguments. In *PRICAI2004 – Proceedings of the Eighth Pacific Rim International Conference on Artificial Intelligence*, 311–321, Auckland, New Zealand, 2004.
2. John Gurney, Don Perlis, and Khemdut Purang. Interpreting presuppositions using active logic: From contexts to utterances. *Computational Intelligence*, 13(3):391–413, 1997.
3. S. J. Kaplan. Cooperative responses from a portable natural language query system. *Artificial Intelligence*, 19:165–187, 1982.
4. Robert E. Mercer. Presuppositions and default reasoning: A study in lexical pragmatics. In J. Pustejovski and S. Bergler, editors, *ACL SIG Workshop on Lexical Semantics and Knowledge Representation (SIGLEX)*, 321–339. 1991.
5. Amihai Motro. SEAVE: a mechanism for verifying user presuppositions in query systems. *ACM Transactions on Information Systems (TOIS)*, 4(4):312–330, 1986.
6. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers, San Mateo, California, 1988.
7. Ingrid Zukerman and Sarah George. A probabilistic approach for argument interpretation. *User Modeling and User-Adapted Interaction, Special Issue on Language-Based Interaction*, 2005.
8. Ingrid Zukerman, Sarah George, and Mark George. Incorporating a user model into an information theoretic framework for argument interpretation. In *UM03 – Proceedings of the Ninth International Conference on User Modeling*, 106–116, Johnstown, Pennsylvania, 2003.
9. Ingrid Zukerman, Michael Niemann, and Sarah George. Improving the presentation of argument interpretations based on user trials. In *AI'04 – Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, 587–598, Cairns, Australia, 2004.