# A Bayesian Approach for User Modeling in Dialogue Systems

AKIBA, Tomoyosi and TANAKA, Hozumi
Department of Computer Science
Tokyo Institute of Technology
2-12-1 Ôokayama Meguro Tokyo 152 Japan
{akiba,tanaka}@cs.titech.ac.jp

## Abstract

User modeling is an important components of dialog systems. Most previous approaches are rule-based methods. In this paper, we propose to represent user models through Bayesian networks. Some advantages of the Bayesian approach over the rule-based approach are as follows. First, rules for updating user models are not necessary because updating is directly performed by the evaluation of the network based on probability theory; this provides us a more formal way of dealing with uncertainties. Second, the Bayesian network provides more detailed information of users' knowledge, because the degree of belief on each concept is provided in terms of probability. We prove these advantages through a preliminary experiment.

## 1 Introduction

Recently many researchers have pointed out that user modeling is important in the study of dialog systems. User modeling does not just render a dialog system more cooperative, but constitutes an indispensable prerequisite for any flexible dialog in a wider domain[9]. The user models interact closely with all other components of the system and often cannot easily be separated from them. For example, the input analysis component refers to the user's knowledge to solve referential ambiguities, and the output generation component does the same for lexical choices.

The concepts are usually explained by showing their relations to the other known concepts. Thus, for the dialog system it is important to guess what the user knows (user's knowledge) in order to explain new concepts in terms of known concepts. For example, consider that the system explains the location of a restaurant to the user. It might be useless to tell the user the position in terms of the absolute coordinate system, since the user's mental model is not based on the absolute coordinate. Therefore, the system should show the relative location from the location the user already knows. It is difficult to predict which locations the user, who perhaps is a stranger to the system, knows. Though the system could attempt to acquire the information by asking the user about her knowledge, too many questions may irritate the user. Such a system is considered mechanical and not helpful. Therefore, the system is required to guess the user's knowledge by finding clues in the user's utterance and to refine the user's model incrementally.

In the user modeling component of UC[5], several stereotyped user models which vary the user's level of expertise were prepared beforehand and the appropriate model was selected based on the user's utterances. In the approach used by Wallis and Shortliffe [12], the expertise level was assigned to all concepts in the user model. The system guessed the user's level, and the concepts with the expertise level lower than her level are considered to be known by her. This model can deal with the level of expertise more appropriately than UC, because the system does not have to prepare the multiple user models for each expertise level.

The approach of preparing several user models and adopting one, however, is an approximation of user modeling. The expertise level of the user is continuous and, in general, the unique measurement of expertise level is not appropriate for some domains, specifically the domain of town guidance considered in this paper, because the areas that are known differ with the users.

Another problem of user modeling is updating the model as the dialog progresses. At the beginning of the dialogue the system cannot expect the user model to be accurate. As the dialogue progresses the system can acquire clues of the user's knowledge from his utterances. Also, the system can assume that the concepts mentioned are known to the user. Thus, updating the user model should be performed incrementally.

One difficulty of updating user models is dealing with uncertainties. The clues that can be obtained from the user's utterances are uncertain, the information may conflict with what has been obtained, and, as a result, the user model may be revised. The effects of the system's explanation are also uncertain. Furthermore, reasoning about the user's knowledge must be performed on the basis of uncertainties. Most previous approaches to this problem are rule-based methods. Cawsey [2] sorted the update rules in order of their reliability and applied them in this order. In another approach, the mechanism such as TMS[6] or nonmonotonic logic[1], is used to maintain the consistency of

the model. It seems that rule-based approaches have a potential defect for dealing with uncertainties[4]. The Bayesian approach can deal with both uncertain (ambiguous) evidences and uncertain reasoning straightforwardly.

In this paper, we propose a probabilistic approach for user modeling in dialog systems. The Bayesian networks are used to represent the user's knowledge and draw inferences from that, and provide the fine-grained solutions to the problems previously mentioned. In spite of the potential advantage of the Bayesian approach, there are few attempts to employ it in user modeling.

The advantages of the Bayesian approach over the rule-based approach are as follows. First, rules for updating user models are not necessary. Cawsey [2] pointed out there are four main sources of information that can be used to update the user model – what the user says and asks, what the system tells the user, the level of expertise of the user, and relationships between concepts in the domain. They can be incorporated in the representation of Bayesian networks and can be used to update the user model by evaluating the networks.

Second, the Bayesian network provides more detailed information of users' knowledge. In the case of binary modeling of knowledge, whereby either the user knows or does not know a concept, it is too coarse to judge the model under uncertainty. Therefore, usually, the degree of belief is assigned to all concepts in the model. It is not clear where the degree of belief comes from or what it means. On the other hand, however, the Bayesian approach provides the degree of belief for clear semantics, which is probability.

The remainder of this paper is organized in four sections. Section 2 is devoted to an outline of Bayesian networks. Section 3, knowledge representation in terms of Bayesian networks is discussed. If the model is once represented, then the updating of the model will be taken care of through the evaluation of the network. Section 4, some examples are given along with an experiment to show the advantage of our approach. Section 5 concludes this paper.

# 2 Bayesian Networks

Reasoning based on probability theory requires probabilistic models to be specified. In general, a complete probabilistic model is specified by the joint probabilities of all random variables in the domain. The problem is that the complete specification of the joint probabilities requires absurd amounts of numbers. For example, consider the case where all random variables are binary, having a value 0 or 1, the complete probabilistic model is specified by $2^n - 1$ joint probabilities. (Assuming $n$ binary random variables, $x_1, x_2, ...x_n$, the distribution is specified by the probabilities, $P(x_1 = 0, x_2 = 0, ..., x_n = 0), P(x_1 = 1, x_2 = 0, ..., x_n = 0), ..., P(x_1 = 1, x_2 = 1, ..., x_n = 1)$, that sum up to unity so one of them can be automatically gained.) Moreover, in practice it is difficult to explicitly specify the joint probability. Concerning our purpose of modeling the user's knowledge, where a random variable corresponds to a concept and whose value corresponds to the user's knowledge of the concept, it is almost impossible to specify all joint probabilities because this involves enumerating all of the user's knowledge patterns.

Bayesian networks need far fewer probabilities and can provide the complete probabilistic models. The information that compensates for the gap is qualitative, which is obtained by investigating the nature of the domain. The Bayesian network has both qualitative and quantitative characteristics, therefore, we can represent the knowledge qualitatively and reason about probability quantitatively. Formally, Bayesian networks are directed acyclic graphs (DAG) with the nodes representing a random variable and the directed arcs representing the direct dependent relation between the linked variables. If a arc goes from one node to another, we say that the former is a parent node of the latter, and the latter is a child of the former. The distribution on the network is specified to all nodes $x$ its probability $P(x|p(x))$ conditioned by the set of its parent nodes $p(x)$. The nodes without parents are assigned the prior probabilities $P(x)$. That is all that is necessary for specifying a complete probabilistic model [10].

The reasoning on Bayesian networks corresponds to evaluating the posterior probability $P(x|E)$ on all nodes $x$ given the evidence $E$ that is specified by providing certain values to a certain subset of nodes in the networks (for instance, $E = \{y = 1, z = 0\}$ for some nodes $y$ and $z$). The evaluation of the network is done in general by the stochastic simulation [10]. The updating of the user models are directly performed by evaluating the network once the knowledge of the domain has been correctly represented by the Bayesian network. In the next section, we discuss knowledge representation with Bayesian networks.

# 3 Knowledge Representation with Bayesian Networks

## 3.1 Designing the Language

We have said the nodes in the Bayesian network are random variables that range over some values. In order to represent knowledge in terms of the Bayesian network, we must design the language for the sentences assigned to the nodes of the network. We first assume that the variables have two possible values, so that the sentences have truth values, that is, 1 (true) or 0 (false). Note that this assumption is not crucial; we may assign values such as KNOWN, NOT-KNOW, NO-INFORMATION as in UMFE [11].

The type of sentences may depend on the application we pursue. For general explanation, it is important to make a clear distinction between the two user's states; knowing the name of a concept and knowing the other attribute of the concept. For example, suppose the user asked the following:

"Where is FRISCO ?"

where FRISCO is the name of a record store. From this question, the system infers that the user knows the name of the store, but does not know its location.

Now we will give a precise definition of our language. All the sentences in the language have the form

⟨ *label* ⟩ : ⟨ *content* ⟩

where ⟨ *label* ⟩ is one of **PRE, POST, JUDGE, TOLD**, and **TELL**, and ⟨ *content* ⟩ is represented by a term of the first-order predicate logic. An object and an expertise field are represented by an atomic symbol, and an attribute of an object is represented by a function symbol. For example, **store001**(object), **records_collector**(expertise field), **location(store001)**(attribute), and so forth.

The user's knowledge about an attribute is represented by five sentences, all having the same ⟨*content*⟩ representing the attribute, and one of the five labels. The sentences labeled **PRE**, express that the user knows the attribute prior to the dialogue session, while those labeled **POST**, express that the user has come to know it during the session. For instance, **PRE: location(store001)** means that the user have already knows the location of **store001** before the interaction starts, while **POST: location(store001)** means the user has come to know the location through the system's explanation. The sentences labeled **JUDGE**, express the user's current knowledge and is used to exploit the user model by other components in the dialogue system. For instance, **JUDGE: location(store001)** means the user now knows the location of **store001**. The sentences labeled **TOLD** and **TELL**, express the evidence, gained by the user's utterance and the system's explanation. For instance, **TOLD: name(store001)** means the user has indicated by the clues that she knows the name of **store001**, while **TELL: name(store001)** means the system has explained the name. For exception, in the case of location, the form **TELL: location(X)**(where X is some object ID) is not used because a location is explained in terms of the relative location of another object. Instead, the form **TELL: relation(X, Y)**(where X and Y are some object IDs) is used.

The sentences representing objects and expertise fields have only the label **PRE**. The sentence representing an object (e.g. **PRE: store001**) means that the user knows the object, that is she knows most of the attributes of the object. The sentence representing an expertise field (e.g. **PRE: records_collector**) means that the user is an expert of the field, that is she knows the objects related to the expertise field.

## 3.2 Constructing the Networks

As mentioned, arcs of the Bayesian network represent direct probablistic influence between linked variables. The directionality of the arcs is essential for representing nontransitive dependencies. In order to represent the knowledge in terms of Bayesian Network, we must interpret the qualitative relation between the sentences that are represented by our language as a directed arc or some such combination of arcs.

In our case, the network has two sub-networks. One represents the user's knowledge before the dialog session, which is used to guess the user's model from her utterances . The sentences assigned to the nodes in this part have either the label **PRE** or **TOLD**. We call this subnetwork the prior part. The other subnetwork in which the nodes have either the label **POST** or **TELL** is used to deal with the influence of the system's utterances. This subnetwork we call the posterior part. It is important to make a clear distinction. Considering that the system explains a concept, it is not proper to assume that the user knows some other related concepts. For example, if the user utters that she knows some location $x$ then it can be inferred that she also knows locations that are close to $x$. But that is not true if the location $x$ is explained by the system.

The relations in the prior part of the network are categorized into four types as follows:

(1) the relations between objects in an expertise field

(2) the relations between attributes of objects

(3) the relations between an object and its attributes

(4) the relations between an attribute of an object and the evidence that the user knows it

The relations (1) are concerned with the expertise field. The objects in the same expertise field are related through the expertise field node. We introduce the arcs that go from the expertise field node to the object nodes belonging to that field. For example, arcs go from the node of "records collector" to that of "Compact Disk","Tower Records" (name of a record store) and so on. The level of expertise can be controlled by the conditional probabilities assigned to the object nodes conditioned by the expertise field node. In this framework, we can introduce arbitrary numbers of expertise fields, all of which can be assigned the level of expertise.

The relations (2) are concerned with the domain knowledge. In our domain, those are the relations between the locations, which are based on the assumption that the user probably knows the locations close to the location she knows. The relations are assumed to be symmetric. A single directed arc of Bayesian networks does not represent a symmetric relation. In order to represent a symmetric relation, we introduce a dummy evidence node, whereby two arcs go forth from the two location nodes as shown in figure 1. The prior

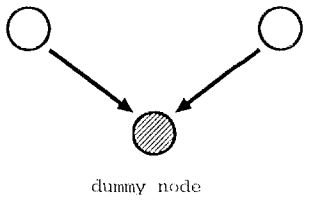


Figure 1: Symmetric relation

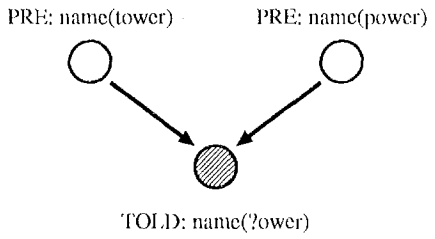


Figure 2: Ambiguous evidence

conditional probabilities of the dummy node have high value if the two parent nodes have the same value.

The relations (3) are concerned with general knowledge, such as knowing an object well implicates knowing its attributes. In order to represent such kind of relations, we introduce the arcs to go from the node of an object to the nodes of its attributes.

The arc corresponding to the relation (4) is introduced to go from the node of an attribute of an object to an evidence node. The attribute node and the evidence node have the same content, while they have the different labels, **PRE** and **TOLD**.

In the posterior part of the network, there are only arcs representing the relations (4). The attribute nodes and the evidence nodes are labeled **POST** and **TELL**. In addition, the **TELL** node may have more than one parent node because the explanations of the attribute are made by referring to the other attributes. Actually, in our town guidance domain, the system explains the new location using the locations that the user already knows. For instance, the nodes **POST: location(store001)** and **POST: location(store002)** are parents of the node **TELL: relation(store001, store002)** when the system explain the location of **store001** by using the location of **store002**. The more the system shows the relations, the deeper the user's understanding becomes.

The ambiguous evidence can be dealt with straightforwardly in the Bayesian approach. An evidence node can have more than one parent node to represent the ambiguity. For example, when dealing with spoken inputs, it might be ambiguous that the user said either "tower records" or "power records." If both record stores exist, an evidence node labeled **TOLD** is introduced as a child node for both nodes, **PRE: name(tower)** and **PRE: name(power)** (figure 2).

Finally, we introduce the arcs that connect the two subnetworks. For each attribute, there are three kinds of nodes labeled **PRE**, **POST**, and **JUDGE**. The two arc are drawn from the **PRE** node to the **JUDGE** node and the **POST** node to the **JUDGE** node. That means the user knows the attribute either because he already knew it before the current dialogue session or because it has been explained by the system during the session.

The example of the resulting network is shown in figure 3.

# 4 Examples

Suppose the user asks the system to show the way to a record store named FRISCO in a town (figure 4). The system uses the network in figure 3. The dialogue starts with the user's request.

(1) user: Where is FRISCO?

In practise, the input analysis component is needed to obtain evidences of the network from the user's utterances, but this process is beyond the scope of this paper. By analyzing the input, the system obtains the information that the user knows the name of a certain store, but does not know its location. The input, i.e. the evidence, to the network is $E$ = {**TOLD: name(frisco)** = 1, **TOLD: location(frisco)** = 0}. Evaluating the degree of belief of each concept $x$ by using the posterior probability $P(x|$ **TOLD: name(frisco)** = 1, **TOLD: location(frisco)** = 0) gives the resulting user model. Though this result can be directly obtained by evaluating the network, we will briefly trace our reasoning for explanatory purposes. (Note that the actual process is not easy to explain as all nodes of the network influence each other, that is the reason why simulation is needed for evaluation.)

The user knows the name FRISCO, which represents that she has the high expertise level for records collectors and raises the probability of the node **PRE: records_collector** and also raises that of the node of other record stores, Tower Records(**PRE: tower**), Wave Records(**PRE: wave**). These nodes then affect the node of their attributes, **PRE: location(tower)**, **PRE: name(tower)**, **PRE: location(wave)**, and so on. That raises the probability of the location node HANDS Department (**PRE: location(hands)**), which is close to the location the user (probably) knows, i.e. **PRE: location(wave)**.

Next, the system generates the answer by using the resulting user model. This task is done by a planner for utterance generation. The system may decide to use the location of HANDS.

(2) system: It is 300m to the south from HANDS Department.

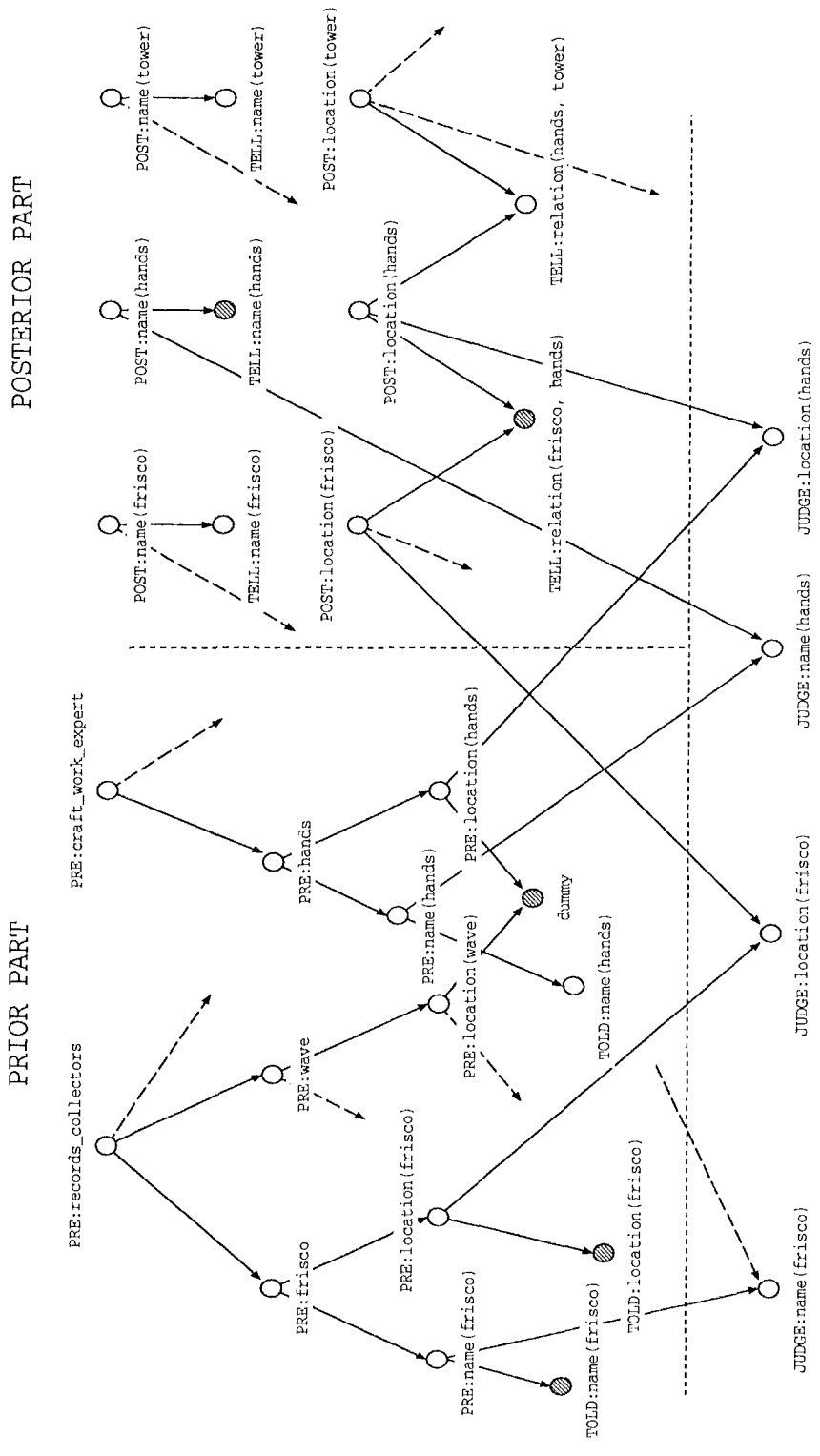


Figure 3: Example of a network

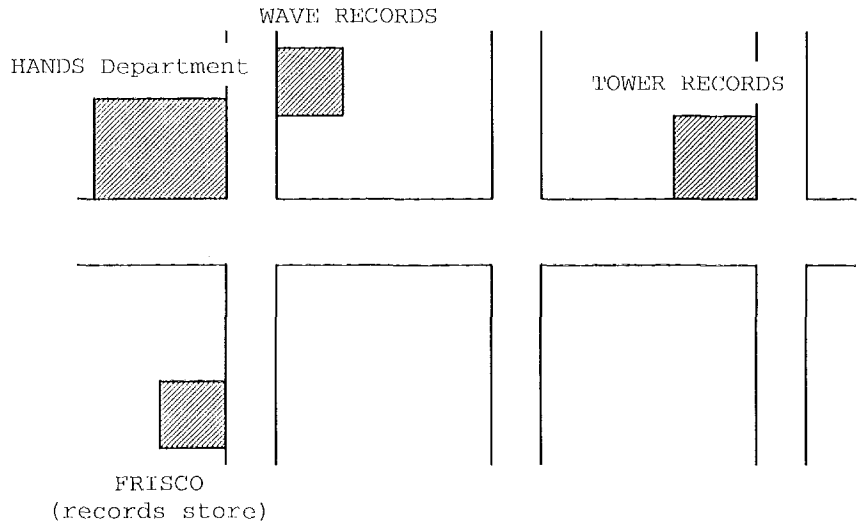


Figure 4: A map of a town

After uttering the sentence, the system adds the evidence, **TELL: name(hands)**= 1, **TELL: relation(hands, frisco)**= 1, to the network. Note that the explanation of the location is made by showing its relation to other locations. That makes the probability of the node **POST: location(frisco)**, $P$(**POST: location(frisco)**$|E$) raise, where $E$ represents all evidence obtained. The next utterance of the user is:

(3) user: I don't know where HANDS is.

This input gives the system the evidence, **TOLD: location(hands)** = 0. After obtaining this evidence, the belief is revised. The probability of the node **PRE: location(hands)** falls, which in turn causes the probability of the node **PRE: location(wave)** to fall.

Next, the planner may try to explain the location of HANDS, by using the location of Tower Records which gives the evidence **TELL: relation(hands,tower)**= 1.

(4) system: HANDS is two blocks away to the west from Tower Records.

This explanation not only can influence the user's understanding of the location of HANDS but also the location of FRISCO, because the evidence raises the posterior probability of the node **POST: location(frisco)** through the node **POST: location(hands)**.

Evaluation results of the above dialogue are shown in Table 1.

# 5 Conclusion

We have proposed the Bayesian approach for user modeling in dialogue systems. The knowledge representation, in terms of Bayesian networks, has been discussed. Reasoning would be automatically and directly performed by evaluating the network followed by stochastic simulation.

Most exact solutions for the interesting problems in artificial intelligence are known to have NP-hard computational complexity. Thus, it has been recognized that solving them by an approximate method is a more realistic approach. The Bayesian networks are evaluated by the stochastic simulation, which is the approximate solution of probabilistic reasoning. The simulation cost, however, is still expensive with the present computing resources. The parallel implementation has reported good performance results [7].

After gaining the accurate expectations of user models, a mechanism to use them for utterance generation is required. This will be done by planners for utterance generation, which try to achieve the system's goals. The probabilities in the user model contribute to measure to what extent the plan will succeed.

In the study of natural language processing, Bayesian approaches have been adopted in the field of plan recognition [3] and lexical disambiguation [7]. We have adopted the Bayesian networks for user modeling because we have perceived that user modeling is one of the core components of dialogue systems whose behavior strongly influences the other parts of the system. We endeavor to construct the experimental dialogue system that accepts the users' inputs by speech recognition[8]. Starting with user modeling, we will ex-

| node | prior | probabilities after the utterance (n) | | | |
|---|---|---|---|---|---|
| | | (1) | (2) | (3) | (4) |
| **JUDGE:location(frisco)** | .51 | .21 | .43 | .43 | .66 |
| **JUDGE:location(wave)** | .48 | .67 | .67 | .31 | .31 |
| **JUDGE:location(tower)** | .51 | .64 | .64 | .58 | .82 |
| **JUDGE:location(hands)** | .48 | .67 | .76 | .43 | .74 |
| **JUDGE:name(frisco)** | .47 | .86 | .86 | .80 | .80 |
| **JUDGE:name(wave)** | .47 | .78 | .77 | .63 | .63 |
| **JUDGE:name(tower)** | .47 | .78 | .77 | .64 | .90 |
| **JUDGE:name(hands)** | .46 | .53 | .87 | .83 | .83 |
| **PRE:records_collector** | .39 | .85 | .84 | .64 | .64 |

Table 1: The result of evaluation

pand the adoption of Bayesian approaches in most of the components in the system. The approaches must be quite effective in the other components, and lead to a system whose components closely interact with each other on the common basis of probability theory.

## References


[1] Douglas E. Appelt and Kurt Konolige. A nonmonotonic logic for reasoning about speech acts and belief revision. In *International Workshop on Nonmonotonic Reasoning*, pp. 164–175, 1988.

[2] A. Cawsey. *Explanation and Interaction.* MIT Press, 1993.

[3] E. Charniak and R.P. Goldman. A bayesian model of plan recognition. *Artificial Intelligence*, Vol. 64, No. 1, pp. 53–79, 1983.

[4] Peter Cheeseman. In defence of probability. In *the Proceedings of the International Joint Conference on Artficial Intelligence*, pp. 1002–1009, 1985.

[5] David N. Chin. KNOME: Modeling what the user knows in UC. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, chapter 4, pp. 74–107. Springer-Verlag, 1989.

[6] J. Doyle. A truth maintenance system. *Artificial Intelligence*, Vol. 12, pp. 231–272, 1979.

[7] Leila M. R. Eizirik, Valmir C. Babosa, and Sueli B. T. Mendes. A bayesian-network approach to lexical disambiguation. *Cognitive Science*, Vol. 17, pp. 257–283, 1993.

[8] K. Itou, S. Hayamizu, and H. Tanaka. Continuous speech recognition by context-dependent phonetic HMM and an efficient algorithm for finding n-best sentence hypotheses. In *In Proceedings of International Coference on Acoustics, Speech, and Signal Processing*, 1992.

[9] A. Kobsa and W. Wahlster, editors. *User Models in Dialog Systems.* Springer Verlag, 1989.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[11] D. Sleeman. UMFE: A user modelling front end subsystem. *International Journal of Man-Machine Studies*, Vol. 23, pp. 71–88, 1985.

[12] J.W. Wallis and E. H. Shortliffe. Customized explanations using causal knowledge. In B.G. Buchanan and E.H. Shortliffe, editors, *Rule Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, pp. 371–390. Addison Wesley, 1985.

# Reserve Papers