A Probabilistic Approach to Text Understanding

Robert P. Goldman
Eugene Charniak

Brown University
Dept. of Computer Science
Providence, Rhode Island 02912

Technical Report No. CS-90-13
May 1990

A Probabilistic Approach to Text Understanding

Robert P. Goldman and Eugene Charniak*
Dept. of Computer Science, Brown University

May, 1990

We discuss a new framework for text understanding. Three major design decisions characterize this approach. First, we take the problem of text understanding to be a particular case of the general problem of abductive inference: reasoning from effects to causes. Second, we use probability theory to handle the uncertainty which arises in abductive inference in general, and natural language understanding in particular. Finally, we treat all aspects of the text understanding problem in a unified way. All aspects of natural language processing are treated in the same framework, allowing us to integrate syntactic, semantic and pragmatic constraints. In order to apply probability theory to this problem, we have developed a probabilistic model of text understanding. To make it practical to use this model, we have devised a way of incrementally constructing and evaluating belief networks which is applicable to other abduction problems. We have written a program, Wimp3, to experiment with this framework.

*This work has been supported in part by the National Science Foundation under grants IST 8416034 and IST 8515005 and Office of Naval Research under grant N00014-79-C-0529.

1 Introduction

For some time we have been interested in the problems posed by uncertainty in text understanding: representing and reasoning about alternative interpretations of NL texts. We are concerned with problems like pronoun reference, prepositional phrase attachment, and lexical ambiguity. We are particularly interested in plan-recognition as it is needed in text understanding: understanding the meanings of stories by understanding the way the actions of characters in the story serve purposes in their plans. Our work builds upon earlier work in script- and plan-based understanding of stories like that of Wilensky [25], Charniak [2] and Norvig [21].

We see text understanding as a particular case of the problem of abduction (reasoning from effects to causes), or diagnosis (see [16] for an outline of this position). In particular, for the case of simple, declarative text, we simplify by viewing the language user as a transducer. The language user observes some thing (event or object) in the 'real world', and translates this thing into language. Our task is to reason from the language to the intentions of the language user and thence to the events described.

We have adopted probability theory as our framework because it is the most thoroughly-understood way of reasoning about uncertainty. Unlike other ways of treating uncertainty (e.g., default logic), probability provides a unified 'currency' in which we can weigh evidence from different sources. Furthermore, it is at least in some sense normative.[8] Finally, techniques for graphically representing probability distributions have made the task of drawing up and reasoning with probabilistic models much less onerous than they were once thought to be.

Our previous work in this area was codified in the program, Wimp2 [3]. Wimp2 used a multiple-context deductive database, based on deKleer's ATMS [12], whose contexts were tagged with probabilities.[1] Having a multiple-context database to cache inferences made it possible to consider simultaneously different interpretations, such as alternate meanings of words, or different parses, and their consequences. The probabilities made it possible to allocate Wimp2's computational resources to considering the most likely candidate explanations.

We were unhappy with Wimp2 for two reasons. First of all, it was difficult to reconcile the logical approach which motivated the use of the ATMS with the use of probability theory. This problem manifested itself particularly acutely in difficulties we had assigning a clear semantics to the probabilities we used. On a more concrete level, it seemed that we were doing the same inference twice over: we were reasoning deductively using the ATMS, and in an ad-hoc probabilistic way when managing the probabilities of the ATMS' contexts. Finally, the TMS framework is unable to distinguish evidential and causal support for propositions. When a new justification is added to a proposition, one cannot tell whether it represents evidence for that proposition, in which case it would confirm hypotheses which explain that proposition; or if it represents an alternative explanation, in which case

[1] At the same time as we developed this database architecture, Laskey and Lehner [19] and D'Ambrosio [10, 11] developed probabilistic ATMS' that were very similar.

it should compete with other explanations. For these reasons, we decided to try a wholly probabilistic approach to the problem.

The third distinguishing feature of our approach is the extent to which it integrates all levels of processing. Previous work in text understanding has almost universally assumed that the text will be pre-processed by an intelligent parser (e.g., [9, 16, 21, 25]). We, on the other hand, read the text word-by-word through a parser that simply returns all possible parses of the input. Because the structure of the sentence is described in the same framework as its semantics and pragmatics, constraints from different levels can be assembled into a global interpretation. For example, in deciding the proper attachment of the prepositional phrase "with some poison" in "Janet killed the boy with some poison." we can make use of our knowledge about murder, or about a particular poison-wielding boy that Janet dislikes.

Our current program, Wimp3, operates by turning problems of text interpretation into probability questions of a form like "What is the probability that this word has this sense, given the evidence?" (where the evidence is the text of the story so far). Wimp3 does this on a word-by-word basis. The probability questions are formulated in a language we have developed for this purpose. This formulation is given additional structure by a graphical representation, belief networks. In the next section of this paper, we will outline the language we have developed to express linguistic problems in terms of random variables. Following that, we will show how these sets of random variables can be structured as belief networks, and how this structure helps us assess the quantities we need for our probabilistic models. Finally, we will show how Wimp3 actually constructs and solves these problems.

2 The Formalism

In changing our approach to a wholly probabilistic one, our first problem was to come up with a clear understanding of what probabilities we were to manipulate. We have created a simplified formal language suited to expressing facts about a text's structure and meaning. We have devised a semantics for this language which allows us to treat its formulae as random variables. We show that this semantics gives us the guidance we need to devise a consistent set of probabilities for use by a story-understanding program. A slightly earlier version of this language is fully reported in [4, 5]; we will only have space for a brief discussion here.

We will start by describing the notation of our language and showing how it is used to formulate the text understanding problem. For convenience's sake, we will proceed in a way which follows the traditional syntax–semantics–pragmatics distinction. First we will show how the language can be used to describe the input text itself. Then we will go on to talk about how it can express semantic propositions. Finally, we will show the part of the language used for plan-recognition, and how this provides contextual information.

The parser and morphological analyzer component of Wimp3 generate the statements that describe the natural-language input. These statements are of two types:

- specifying the words used (*word-inst*), and

- specifying relations between words used (*syn-rel*).

For example, the sentence "Jack gave Mary the ball." (1) would be translated into:

(word-inst word1 Jack)
(word-inst word2 give)
(syn-rel subject word1 word2)
(word-inst word3 Mary)
(syn-rel indirect-object word3 word2)
(word-inst word4 ball)
(syn-rel object word4 word2)
(syn-rel det word4 the)

In the case of syntactic ambiguity, the parser will generate and describe all possible parses of the input; semantic and pragmatic information will be used to choose between them. If the sentence were "Mary killed the boy with the poison." (2), the parser would give both (syn-rel with poison12 boy11) and (syn-rel with poison12 kill10).[2]

For semantic processing, we need to be able to express hypotheses about the denotations of the various words. Associated with each word instance is a constant representing the denotation of that word. We will write this as the same constant, but in boldface. We might more clearly write (denotation word$n$), but this is too bulky. For example, if ball27 was the constant representing the denotation of 'ball' in (1) above, two possible denotations might be expressed as:

(inst ball27 bouncing-ball)
(inst ball27 cotillion)

In order to express case relations, we have functions for the various cases. To return to example (2) ("Mary killed the boy with the poison."), the case relations corresponding to the different prepositional phrase attachments would be:

(== (accompaniment boy11) poison12)
(== (instrument kill10) poison12)

Moving into the gray area between semantics and pragmatics that Hobbs and Martin [15] call 'local pragmatics,' equality statements are used to express coreference. For example, part of the task of understanding "Jack gave Mary the ball. She liked it." (3) is to reason from

(inst mary24 girl-)
(inst she28 girl-)

[2]For ease of understanding we will use these more descriptive names, rather than word1, word2, etc.

to (== she28 mary24).

Finally, we need to be able to express a plan-based understanding of our texts, such that when reading a story like "Bill went to the liquor-store. He paid for some bourbon." (4) our program can understand the entire story as a description of a plan to buy liquor. Our database records what kinds of plans can explain each action or object.[3] E.g., a going action can be the go-step of a liquor-shopping the go-step of a robbery, etc. For this purpose we have another kind of proposition: the proposition that a given frame-instance (say went2) fills a slot in a given kind of plan (say robbery). This would be written

(and (inst *robbery27* robbery)
(== (go-step *robbery27*) went2)) (5)

The atom *robbery27* is written in a different typeface to indicate that it represents a different kind of constant. *robbery27* is a function of went2. Analogously to the the denotation constants, these constants might more clearly be written (robbery-explanation-of went2). Put differently, the formula given above as (5) is akin to an existentially quantified formula whose variable is *robbery27* and whose scope is contained within that of went2.

These propositions are virtually identical to those in our earlier logic for semantic interpretation [3], but their semantics is different. Following probability theory, the expressions of our language get their semantics from a sample space. This sample space is the set of all possible stories from some restricted universe. The word constants denote random variables whose domain is particular words. Likewise, the denotations of words are random variables whose domain is the set of entities (including events). Propositions are random variables whose domain is the set {true,false}. A more full discussion of the semantics of our language can be found in [5].

3 Network representations

In the past, it has been difficult to apply probability theory to reasoning problems because such problems did not have sufficient structure. It appeared that in order to apply probability theory, it would be necessary to have a gigantic joint probability table, giving the probability of all possible combinations of the propositions of interest. However, recently, there has been renewed interest in graphical representations for probability distributions, such as markov fields, belief networks and influence diagrams. Belief networks have provided us with a way of structuring the problem of text understanding as a problem of probabilistic inference, and have simplified the problem of acquiring the requisite numbers.

Belief networks are a way of representing probabilistic dependency information in directed acyclic graphs. Such graphical representations have been used in various fields, for some time (Lauritzen and Spiegelhalter [20] cite uses in path analysis as early as 1921). These graphs make possible qualitative reasoning about influence, and for distributions

[3]We have a frame-based database of events and objects.

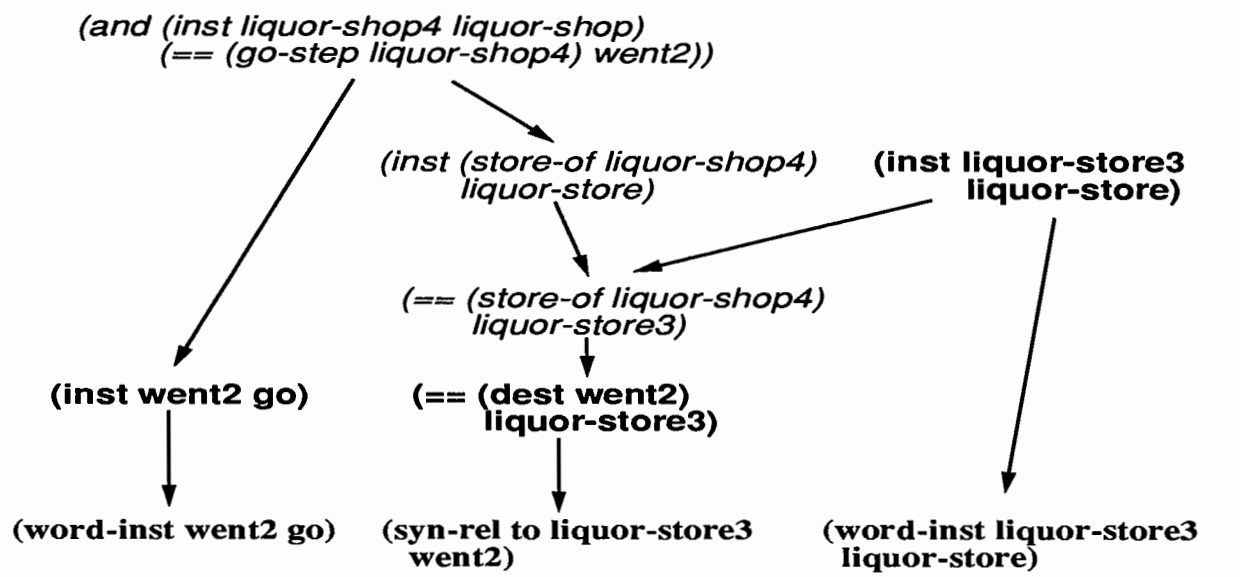

Figure 1: A belief network representation of the story "Jack went to the liquor-store...."

with desirable properties of conditional independence, make it possible to efficiently calculate posterior probabilities. Judea Pearl's book [22] gives a thorough account of the properties of such networks.

The nodes in a belief network represent random variables, and the edges represent direct dependence. We follow Pearl's suggestion that edge direction be assigned causally: nodes at the tails of edges should be (direct) causes of the nodes at their heads. There are three advantages to belief networks as representations for probability distributions. First of all, properties of conditional independence can be read off a belief network. Second, the probability distribution corresponding to a belief network may be represented locally. For each node, it suffices to provide a conditional probability distribution for each combination of values of its parent nodes. Finally, while in general the problem of determining the posterior distribution of a partially-instantiated belief network is NP-hard [7], considerable attention has been devoted to finding efficient approaches to evaluating such networks.

We give a partial belief network representation of the analysis of example the sentence "Bill went to the liquor-store." (6) in Figure 1. This fraction of the network is intended to capture the following relations: A liquor-shopping plan can cause there to be a go action. It will also cause there to be a liquor-store, the store-of the liquor-shopping. The liquor-store mentioned in the story, might be this liquor-store. The store-of the liquor-shopping will be the destination of the go-step of the liquor-shopping. These relations might be realized in the text sequence we have observed.

There are two things to be noted about this diagram. First of all, the 'explanatory' variables (e.g., liquor-shop4) are tied to the actions of characters, rather than to all entities mentioned. The liquor-store simply supports the liquor-shop hypothesis, and that only in the presence of evidence for it being the destination of Jack's going.[4] This avoids having

[4]This follows from the properties of the belief network representation.

objects call into consideration every plan in which they could play a role. For example, we don't consider a shopping explanation when we read "Jack bombed the supermarket."

Note also that this is an immensely simplified version of the networks which are, in fact, used in our program. First of all, we have suppressed many of the nodes that would appear in the network for this interpretation (e.g., the nodes specifying Jack as the agent of the go). Second, the networks we construct are not restricted to contain only nodes which are part of *correct* interpretations of the text!

4 Quantifying the networks

The belief network representation makes it easier for us to specify the probability distributions we need. It permits us to express the distribution as a collection of conditional probabilities involving only a few nodes (no more than 3 in our current program). This representation is compact, and these low-order probabilities are easier to assess subjectively.[5] In this section we will give a brief discussion of the kinds of conditional probabilities we need For the sake of clarity, we will proceed from the leaves of the belief networks to their roots, in parallel with the syntax–semantics–pragmatics distinction, and our treatment in section 2.

The leaves of the network are the word-inst and syn-rel nodes. For this portion of the graph, the belief network formalism requires us to provide P(denotation) [P(d)] and P(word|denotation) [P($w|d$)]. In principle, these could be computed from a labelled corpus. In practice, this is not necessary. Because the denotation variable is a function of the word variable, its probability is zero in the absence of the word. Accordingly, we are only concerned with P($d|w$). We can specify these by ranking the various different denotation hypotheses relative to each other. Precise values of P(d) and P($w|d$) are not important: it is enough to know the values of the products P(d)P($w|d$) *relative to other possible denotations for the same word.*[6]

We do not have a good theory of the conditional probabilities for the syn-rel nodes, which are of the form P(syn-rel|relation in the real world). E.g., P((syn-rel to store2 go1))|(== (destination go1) store2)). We use a rough subjective estimate of how often a given construction will be used to convey some relation. E.g., the instrument relation is more often expressed by 'with' phrases than is the accompaniment relation (which is often expressed by conjunction).

Deeper in the graph, we need probabilities for propositions which represent explanatory hypotheses. For example, the hypothesis that Jack has a shopping plan which explains his going to the liquor-store. This is the node labelled (and (inst *liquor-shop4* liquor-shop) (== (go-step *liquor-shop4*) went2)) in Figure 1. (I will refer to this explanatory node as e, for the sake of brevity.)

[5]While in principle, we could collect statistics to find these distributions, this is not practical.

[6]The exact values of the products are not necessary since they will be normalized. This also enables us to do without P(w).

The task of quantifying this part of diagram may seem well-nigh impossible, because it appears to require that we specify a prior probability for this node. In fact, however, this is not the case. This part of the network is subject to the same analysis as that of the words and denotations. Again, this is because the explanatory nodes have meaning only in the presence of their explanands. So we can specify this part of our network by ranking the different explanations for each action relative to each other.

In our program, we have tried to tie these probabilities to the frequencies of events in the real world. We want to do this so that, e.g., when our program reads a sentence like "Mary went to the airport." it will favor explanations like 'Mary is going to meet someone on the plane,' or 'Mary will fly somewhere,' over 'Mary is going to hijack the plane.' This is because we're assuming the speaker is a transducer. In theory, one could collect statistics for these networks by labelling a large corpus of stories, and counting nodes in the instantiated networks.

Finally, we need probabilities of the form P((== thing1 thing2)|(inst thing1 type1), (inst thing2 type1)). These are needed for coreference and for linking together explanations for different objects (we must recognize that the shopping plan that explains the presence of a liquor-store in a story is the same plan that explains the presence of a bottle of bourbon). These are difficult to assess; [4] discusses the problem of assessing such probabilities. We do not have time to review the results fully here. In brief, these probabilities must be sensitive to the number of different objects of each type are likely to appear in a given story. For example, in simple stories like those Wimp3 reads,

P((== person1 person2)|(inst person1 person), (inst person2 person))
  < P((== airport1 airport2)|(inst airport1 airport), (inst airport2 airport))

because it is very likely that there will be more than one person discussed in a single story, but unlikely to be more than one airport. We hope that discourse theory (e.g., [14, 24]) will shed some light on these quantities, and that notions like focus can be used as conditioning events.

5 The Program

In the preceding, we have developed our probabilistic model of stories. Now we will show how we use this model to do story understanding. Because the network models we have outlined here are, in theory, infinite, and different for each input, we proceed by incrementally constructing and evaluating relevant portions of the full network. These partial networks are constructed by network-building rules which are similar to forward-chaining rules. We have experimented with evaluating them using a number of different belief network algorithms.

Wimp3 works as follows: an all-paths parser reads the English-language input, one word at a time, and yields statements about the input. These statements are given to the network-construction component. This component extends the belief network corresponding to the input. Then the network evaluation component computes the posterior

```
(→(word-inst ?i ?word) :label ?A
   (→←(word-sense ?word ?frame) :label ?B
         (inst ?i ?frame) :label ?C)
   :prob ((?C →?A) ((t | t = ?B) (t | f = prior)) ))
```

Figure 2: One of the rules used to handle word-sense relationships.

distribution of the network. If any hypotheses seem particularly good or bad, they may be accepted or rejected categorically. After this, if necessary, the network may be further extended, and re-evaluated. At this point, control is returned to the parser, which reads the next word.

We have developed network building rules for constructing belief networks which correspond to story understanding problems.[7] These rules are similar to the forward-chaining rules in a conventional TMS. They differ in that rules are not restricted to adding justifications to some derived statement. In fact, since we are trying to build networks for a diagnostic problem, we will usually be extending our networks *above* the triggering node: i.e., adding possible causes for this statement. To put it differently, our rules will typically be triggered by the heads of arcs they add, rather than by the tails. Our network-building rules also provide more information than simple connectivity: they contain information used to compute the conditional probability matrices of the nodes in the network.

A sample network-building rule is given in Figure 2. This rule handles part of the problem of finding an appropriate referent for a word (the rules for plan-recognition are similar, but more complex). The meaning of the rule is:

> If a node is added to the network describing a new word-token, and if there is a statement in the database specifying that one of the senses of this word is ?frame, add a node for an instance of the type ?frame. Draw an arc between this newly-added node (?C), and the word-inst node (?A). Construct the conditional probability matrix for the word-inst node using information from the word-sense statement (?B).

For example, if we were to learn of a use of the word 'bank,' (say bank1) and knew of two meanings: financial institution and river bank, this rule would tell us to add two nodes to our network: a node specifying that bank1 referred to a financial institution, and a node specifying that bank1 referred to a river bank. These two nodes would have arcs pointing to the node for (word-inst bank1 bank). The conditional probability matrix for the latter would be drawn up to reflect that the two possible senses of the word are related as exclusive causes. I.e., the probability that the word will be used given that one wishes to describe a financial institution or a river bank is some value determined by the word-sense rule; the probability that the word is used given one wishes to express both

[7] See also [13].

senses simultaneously is zero, and the probability that the word will be used given neither of these senses is meant is some default value.

We have similar rules for syntactic relations, reference resolution, etc. Initially we used this kind of 'forward-chaining' for plan-recognition, as well. However, this led to an explosive growth in the belief networks. The reason for this is simple: consider how many possible reasons there are for 'going,' for example. To provide more direction to the way the networks are expanded, another researcher at Brown, Glenn Carroll, has developed a marker-passing search algorithm. For a discussion of the marker-passer, see [1].

The posterior distribution for the belief network constitutes the interpretation of the input text. To see why this is the case, refer to the sample belief network given as Figure 1. What we want to assess is the probability that Jack is going to buy liquor, given that we have been told "Jack went to the liquor-store." We can read this information off the network, once we have evaluated it so that it reflects the posterior distribution – the probability of each node given the evidence in the story. It should be noted that our representation does not tie us to the idea of solution by computing posterior distributions: we could compute a maximum a posteriori instantiation of the networks to get a best global interpretation. We do not do this because we want intermediate results, and because we do not know of efficient MAP algorithms that work with distributions which are not strictly positive.

We have been experimenting with a number of different algorithms for computing the posterior distribution of these networks.[8] We find that a variant of the algorithm of Lauritzen and Spiegelhalter[20] developed by Jensen, et. al.[17], gives the best results on networks like ours.[9] Simulation-based approaches don't work on networks which contain extreme probabilities[6], and our cutsets grow too large for conditioning[22].

Wimp3 is has been fully implemented. It is written in Common lisp, and runs on Symbolics Lisp Machines and Sun SPARCstations.

6 Summary

The contributions of this work are two-fold. First of all, we have constructed a probabilistic model of story comprehension. This makes it possible to address the problem of text understanding within axiomatic probability theory. Second, we have developed a way of constructing and evaluating probabilistic models on an as-needed basis. This makes it possible to apply probability theory to problems for which a precompiled model is either not available, or impractically large.

[8] We have been using the IDEAL system[23], an environment in which one can define a belief net or influence diagram and apply to it a number of different evaluation algorithms.

[9] We also make use of a clustering heuristic due to Kjærulff.[18].

References

[1] Carroll, Glenn and Charniak, Eugene, Finding Plans with a Marker-Passer, *Proceedings of the Plan-recognition workshop*, 1989.

[2] Charniak, Eugene, A neat theory of marker passing, *Proceedings of the Sixth National Conference on Artificial Intelligence*, 1986, 584–589.

[3] Charniak, Eugene and Goldman, Robert P., A logic for semantic interpretation, *Proceedings of the Annual Meeting of the ACL*, 1988.

[4] Charniak, Eugene and Goldman, Robert P., Plan Recognition in Stories and in Life, *Proceedings of the Workshop on Uncertainty and Probability in AI*, Morgan Kaufmann Publishers, Inc., 1989.

[5] Charniak, Eugene and Goldman, Robert P., A semantics for probabilistic quantifier-free first-order languages, with particular application to story understanding, *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 1989.

[6] Chin, Homer L. and Cooper, Gregory F., Stochastic simulation of bayesian belief networks, *Proceedings of the Workshop on Uncertainty and Probability in Artificial Intelligence*, 1987, 106–113.

[7] Cooper, Gregory F., *Probabilistic inference using belief networks is NP-hard*, Technical Report KSL-87-27, Medical Computer Science Group, Stanford University, 1987.

[8] Cox, R.T., Probability, frequency and reasonable expectation, *American Journal of Physics*, **14** (1946) 1–13.

[9] Cullingford, Richard E., *Script Application: Computer Understanding of Newspaper Stories*, Technical report, Yale University Department of Computer Science, 1978.

[10] D'Ambrosio, Bruce, A Hybrid approach to reasoning under uncertainty, *International Journal of Approximate Reasoning*, **2** (1988) 29–45.

[11] D'Ambrosio, Bruce, Process, structure and modularity in reasoning with uncertainty, *The Fourth Workshop on Uncertainty in Artificial Intelligence*, 1988, 64–72.

[12] deKleer, Johan, An assumption-based TMS, *Artificial Intelligence*, **28** (1986) 127–162.

[13] Goldman, Robert and Charniak, Eugene, Dynamic Construction of Belief Networks, To appear in the proceedings of the 1990 Conference on Uncertainty in Artificial Intelligence, 1990.

[14] Grosz, Barbara J. and Sidner, Candace, Attention, Intention and the Structure of Discourse, *Computational Linguistics*, **12** (1986).

[15] Hobbs, Jerry R. and Martin, Paul, Local Pragmatics, *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, 1987, 520–523.

[16] Hobbs, Jerry R., Stickel, Mark, Martin, Paul, and Edwards, Douglas, Interpretation as Abduction, *Proceedings of the 26th Annual Meeting of the ACL*, 1988, 95–103.

[17] Jensen, Finn V., *Bayesian Updating in Recursive Graphical Models by Local Computations*, Technical Report R 89-15, Institute for Electronic Systems, Department of Mathematics and Computer Science, University of Aalborg, 1989.

[18] Kjærulff, Uffe, *Triangulations of graphs – algorithms giving small total clique size*, Technical Report R 90-09, Institute for Electronic Systems, Department of Mathematics and Computer Science, University of Aalborg, 1990.

[19] Laskey, Kathryn B. and Lehner, Paul E., Belief Maintenance: An integrated approach to uncertainty management, *Proceedings of the Eighth National Conference on Artificial Intelligence*, 1988, 210–214.

[20] Lauritzen, S.L. and Spiegelhalter, David J., Local Computations with probabilities on Graphical Structures and their Application to Expert Systems, *Journal of the Royal Statistical Society*, **50** (1988) 157–224.

[21] Norvig, Peter, Inference in Text Understanding, *Proceedings of the Seventh National Conference on Artificial Intelligence*, 1987, 561–565.

[22] Pearl, Judea, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, (Morgan Kaufmann Publishers, Inc., 95 First Street, Los Altos, CA 94022, 1988).

[23] Srinivas, Sampath and Breese, Jack, *IDEAL: Influence Diagram Evaluation and Analysis in Lisp.*Rockwell International Science Center, May 1989.

[24] Webber, Bonnie Lynn, The interpretation of tense in discourse, *Proceedings of the 25th Annual Meeting of the ACL*, 1987.

[25] Wilensky, Robert, *Planning and Understanding*, (Addison-Wesley, Reading, Mass., 1983).