ISCA Archive
http://www.isca-speech.org/archive

5th International Conference on Spoken Language Processing (ICSLP 98)
Sydney, Australia
November 30 - December 4, 1998

SPOKEN LANGUAGE UNDERSTANDING WITHIN DIALOGS USING A GRAPHICAL MODEL OF TASK STRUCTURE

Jerry Wright, Allen Gorin and Alicia Abella

AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932, USA
{jwright, algor, abella}@research.att.com

10.21437/ICSLP.1998-507

ABSTRACT

We describe a procedure for contextual interpretation of spoken sentences within dialogs. Task structure is represented in a graphical form, enabling the interpreter algorithm to be efficient and task-independent. Recognized spoken input may consist either of a single sentence with utterance-verification scores, or of a word lattice with arc weights. A confidence model is used throughout and all inferences are probability-weighted. The interpretation consists of a probability for each class and for each auxiliary information label needed for task completion. Anaphoric references are permitted.

1. INTRODUCTION

We are interested in spoken dialog systems in which the caller responds using fluent natural language to the prompt "How may I help you?" (HMIHY). In previous work we have described the speech recognizer [1], automatic acquisition of salient phrase and grammar fragments, and call-type classification [2,3], the dialog manager [4], and the incorporation of utterance verification [5]. In this paper we consider the issue of spoken language understanding within dialog.

At each turn in the dialog, the speech-recognized response from the caller must be interpreted, see [6-10]. Ambiguities and conflicting information need to be resolved so far as possible using confidence scores and dialog context, and to the extent that this is not possible the dialog manager may initiate a clarification sub-dialog [4]. Auxiliary information that is necessary for completing a service must be extracted. Furthermore, a range of proficiency from novice to power-user must be catered for.

In this paper we describe an approach to context-dependent interpretation in which task structure is represented in a simple graphical form. The block diagram of the understanding system is shown in figure 1. The latest recognized sentence with utterance-verification scores, or alternatively a word lattice with arc weights, is initially passed through a task-dependent preprocessor in order to replace certain classes of words or word-strings (such as pronouns or digit strings) with nonterminal symbols. The understanding module itself is task-independent, and consists of a classifier and an interpreter. The purpose of the classifier is to find the surface meaning of the latest sentence, and that of the interpreter is to follow the implications of this for the dialog context.

The interpreter algorithm performs an inference using the task structure graph, the dialog context and the classification of the sentence, in order to arrive at a result which is conveyed to the dialog manager. The dialog manager makes use of the outcome to determine the status of the dialog after the latest turn, and can then initiate the next turn accordingly.

The task structure graph governs the behaviour of both the dialog manager and the understanding module. It also serves as the basis for the protocol for exchanging information between the two modules. The dialog context is sent from the dialog manager to the understanding module, expressed as a path into the graph, and the result returned to the dialog manager consists of a set of probabilities for the nodes of the graph. In this paper we describe the graph structure, the interpreter algorithm, and application to two dialog tasks.

2. TASK STRUCTURE GRAPH

The task structure is specified using three small data files, which together define the graph. This has a dual rôle in our system: it represents both the semantic structure including the needed secondary attributes (similar to the E-form in [7]), and the sources of ambiguity that need resolution. The graph nodes represent the following:

- call-type labels: the set of services that characterize the application, and
- auxiliary labels: the secondary attributes necessary for completing the service.

The connections within the graph are of three kinds:

- primary arcs: directed from node to node to represent the *is-a* and *has-a* relations between labels [10],

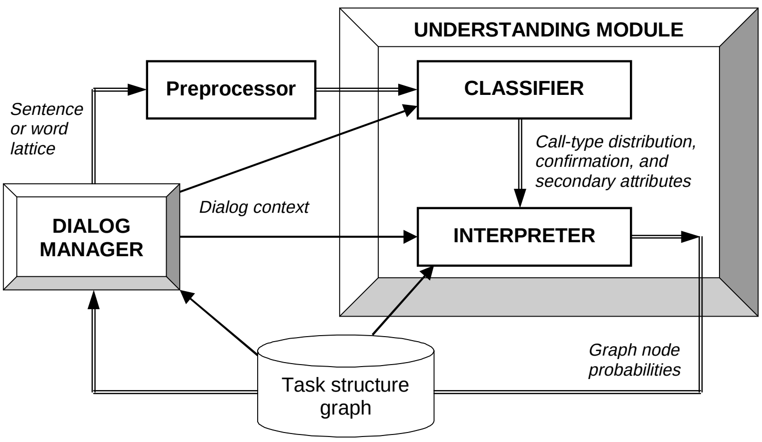

Figure 1: Block diagram of spoken language understanding system.

- exclusives: undirected, expressing the in-compatibility of certain sets of labels,
- implicit confirmations: directed (possibly bi-directional), expressing the ways in which one label may implicitly confirm another.

The graph should not be interpreted as a Bayesian network: absence of an arc does not indicate independence. The graph for a customer-services application [3] is shown in Figure 2. The chain-dot arc from "forward_number" to "collect" is an example of an implicit confirmation; most of these are omitted for simplicity.

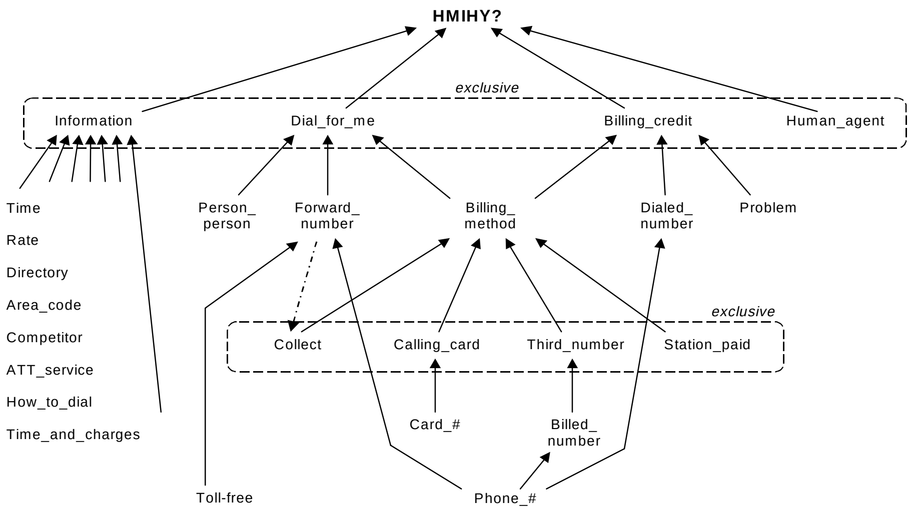

Figure 2: Task-structure graph for a customer-services application.

3. INTERPRETER ALGORITHM

3.1 Summary

The interpreter uses a three-pass algorithm over the nodes in the graph. The dialog context is received from the dialog manager and is expressed as a path through the graph to the apex. For example, the following path in Figure 2:

calling_card→billing_method→dial_for_me→HMIHY?

applies in calling_card situations where the dialog manager is seeking confirmation: "Do you want to make a card call?". The apex node can be omitted. The lower end of the path (first node) is the focus for the latest turn, and normally any explicit confirmation or denial refers to that focus. The only implicit confirmation arcs that are active terminate at the focus. An important step in the interpretation arrives at a decision for the status of the focus after the latest sentence from the user. During the first pass, information is propagated through the graph so that evidence supporting and opposing the focus is gathered. Opposing evidence may consist of an explicit denial, or of supporting evidence for a node incompatible with the focus. A final probability for the focus is assigned, which may involve reconciling the evidence in favour with that against it. A final probability for all other nodes is arrived at through the second and third passes. The order of visiting the nodes is different for each pass, and is based on a topological sort using all the arcs that are active for that pass.

3.2 Forms of implication

The basic principle for propagation of evidence is the rule

$$\text{If } X \to Y \text{ then } P(Y) \geq P(X)$$

This is used in three ways:

- Forward implication (up the graph along primary and implicit confirmation arcs),
- Backward implication (down the graph) in the form: If $\overline{Y} \to \overline{X}$ then $P(X) \leq P(Y)$,
- Transverse implication (across the graph) in the form: If $X \perp Y$ then $P(X) \leq P(\overline{Y})$.

(The notation $X \perp Y$ denotes that *X* and *Y* are incompatible, see below for definition).

In order to propagate evidence up the graph from a node, a probability distribution is assigned over the destination nodes for the arcs leaving that node. Nodes are considered in topological order and evidence is gathered at each node (*Z*) using a simple disjunctive rule

$$P_{\text{forward}}(Z) = \max\left\{P_{\text{direct}}(Z), \max_{U} P(Z \mid U) P_{\text{forward}}(U)\right\}$$

Alternative disjunctive combination rules could be used, but have not yet been tried. The direct evidence ($P_{\text{direct}}(Z)$) arrives from the classifier. For the inner maximization, all propagation probabilities are zero except for the child nodes *U* of *Z*, and any node *U* for which all paths up the graph pass through *Z*, in which case the propagation probability is unity. The distribution from *Z* over its parent nodes is computed after this rule is applied, because it may depend upon the outcome at *Z*.

Transverse implication involves a conjunctive rule:

$$P_{\text{transverse}}(Z) = \min\left\{P_{\text{forward}}(Z), \min_{U:U \perp Z} P_{\text{forward}}(\overline{U})\right\}$$

For any node *Z*, the set of nodes *U* that are incompatible with it is in general larger than the initial set of exclusives given as part of the graph specification (section 2). For any two nodes *U*,*V*, if there exist disjoint subsets S_U, S_V of nodes such that all paths from *U* pass through S_U , all paths from *V* pass through S_V , and each element of S_U is exclusive of each element of S_V , then *U* and *V* are incompatible: $U \perp V$. The set of all incompatible pairs of nodes is computed in advance, from the graph specification tables.

Backward implication also involves a conjunctive rule:

$$P_{\text{backward}}(Z) = \min\{P_{\text{forward}}(Z), P_{\text{backward}}(\tilde{p}(Z))\}$$

where $\tilde{p}(Z)$ is the first node reached from Z (in topological order) such that all paths from Z pass through it. Every node in the graph, except the apex node, has such a point, and this is important for both forward and backward implication:

$$P(\tilde{p}(Z) \mid Z) = 1$$

These inference rules are consistent with various logic systems including the many-valued logic of Łukasiewicz [11].

3.3 Propagation probability assignment

A node may have several parent nodes, representing a situation that is (by itself) ambiguous. This situation is reflected in the interpreter by the probability distribution over the parent nodes. Various factors can influence this:

1. Training: given sufficient annotated dialogs, the distribution for each node in each context could be induced.
2. Default [7]: one parent is nominated as the default unless over-ridden by factors 3 to 5; this is the approach currently implemented.
3. Dialog context: the path into the graph expresses a dialog expectation, and directing the flow of evidence (via the distributions) along paths that intercept the context path as low as possible minimizes the scope of context switches.
4. Other evidence: evidence reaching two or more exclusive nodes constitutes an inconsistency, so the distribution assignment is designed to minimize this.
5. Bounded parents: during the second pass of the algorithm, if a final decision concerning a context path node can bound the propagation to certain parents of a node (by backward propagation) then the distribution is re-assigned to reflect this.

In fact we use a combination of factors 3 to 5 to over-ride the default, with factor 5 having highest priority, followed by 3 followed by 4. This is the most complex aspect of the algorithm.

3.4 Description of algorithm

During the first pass, all evidence that may be interpreted as supporting the dialog focus is allowed to do so. This is ensured by factor 3 in section 3.3, because the focus has greatest depth. All other evidence is propagated in accordance with factors 2 to 4, using forward propagation only. The overall evidence supporting the dialog focus is then combined with that for explicit confirmation (if any) and set against any contrary evidence either from explicit denial or from incompatible nodes by transverse implication. This involves models using both disjunctive and conjunctive combination, and finding a good model is a largely heuristic matter.

During the second pass, supporting evidence for each node (other than the dialog focus) is pulled in from its child nodes and set against contrary evidence from incompatible nodes. Precise details depend on whether the node lies on or off the context path, and again finding a good model is largely heuristic. The result is a final probability for a node which is on the path, but is another intermediate result for a node which is off the path. Application of factor 5 in section 3.3 requires careful control over the order of visiting the nodes, and this is different from that in the first pass.

During the third pass, the nodes are visited in reverse topological order and backward propagation is applied in order to bound the final result for off-context-path nodes.

4. APPLICATIONS

4.1 Customer services

To illustrate the operation of this procedure, consider the customer-services application [3] with the task structure graph of Figure 2. Services offered include call-completions ("dial_for_me") and billing credits, with various billing methods possible in either case. The following examples illustrate two of the behaviour modes.

Evidence over-rides default
Context path: HMIHY? (apex node)
Prompt: "How may I help you?"
Recognition: "I dialed <phone_#> and got a wrong number"
Interpretation: phone_#→dialed_number→billing_credit
problem↗

The nonterminal symbol "<phone_#>" is inserted by the preprocessor. The interpretation should be viewed in conjunction with the graph. The evidence from the billing_credit problem over-rides the default interpretation of a phone number (as the forward number for a call-completion), and it then becomes the "dialed_number" for a billing credit request.

Context resolves ambiguity
Context path: collect→billing_method→dial_for_me
Prompt: "Do you want to make a collect call?"
Recognition: "No to my card number <card_#>"
Interpretation:
card_#→calling_card→billing_method→dial_for_me

Context path: station_paid→billing_method→billing_credit
Prompt: "Was the call billed to the phone you're calling from now?"
Recognition: "No to my card number <card_#>"
Interpretation:
card_#→calling_card→billing_method→billing_credit

The same recognized sentence in each case, changes the billing method but confirms the high-level service: "dial_for_me" or "billing_credit" respectively.

When regarding these illustrations it is important to bear in mind the confidence model. The input from the recognizer has utterance-verification scores or lattice arc weights, the output to the dialog manager is a probability vector, and all the reasoning in between maintains a continuity from the one to the other. If the recognized sentence has a high score (in the given situation) then the nodes listed in the interpretation will also receive a high score.

Figure 3 shows ROC curves for a test set of 1841 spoken responses to the prompt "How would you like to bill this call?", which occurs within call-completion dialogs, (part of a data-collection exercise in November 1997 involving 7744 telephone dialogs). This illustrates context-dependent interpretation, and the context path in this case is billing_method→dial_for_me. The results show that the algorithm described in this paper out-

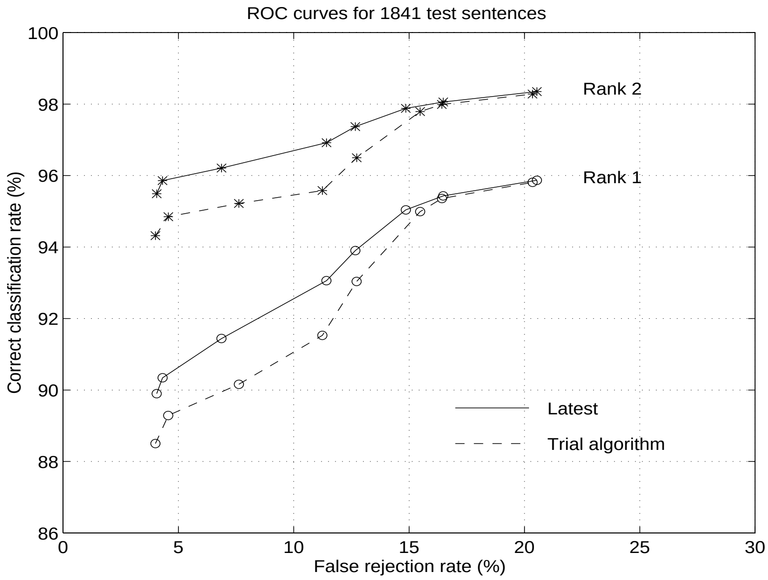

Figure 3: ROC curves for “billing method” response sentences.

performs an earlier version which was deployed for that trial, which incorporated a much simpler task structure and inference model.

4.2 VPQ: a directory information service

For a second application see Figure 4: the task structure graph for VPQ, a voice-activated directory information service [12]. Through this structure the user may call or page individuals identified by name, or engage in a dialog in order to search for some information. The dialog may continue through a series of services within a single session. The implicit confirmation arc from “action” to “continue” allows a user to respond affirmatively to the question of whether they require a further service by actually initiating one. Anaphoric references to individuals are also permitted.

5. CONCLUSIONS

We have described a procedure for contextual interpretation of spoken sentences within dialogs, in which a probability-weighted inference is performed using a graphical model of task structure. Evidence is propagated between the nodes of the graph in a way controlled by the dialog context, and is combined at the nodes using logical rules. The procedure delivers a result to the dialog manager from which the future direction of the dialog can be derived. The procedure has so far been applied to two applications.

6. REFERENCES

1. Riccardi,G., Gorin,A.L., Ljolje,A. and Riley,M., "A spoken language system for automated call routing", *Proc. ICASSP, Munich 1997, pp.1143-1146.*
2. Wright,J.H., Gorin,A.L. and Riccardi,G., "Automatic acquisition of salient grammar fragments for call-type classification", *Proc. Eurospeech, Rhodes 1997, pp. 1419-1422.*
3. Gorin,A.L., Riccardi,G. and Wright,J.H., "How may I help you?", *Speech Communication, vol 23 (1997) pp. 113-127.*
4. Abella,A and Gorin,A.L., "Generating semantically consistent inputs to a dialog manager", *Proc. Eurospeech, Rhodes 1997, pp.1879-1882.*
5. Rose,R.C., Yao,H., Riccardi,G. and Wright,J.H., "Integration of utterance verification with statistical language modelling and spoken language understanding", *Proc. ICASSP, Seattle 1998.*
6. Seneff,S., Goddeau,D., Pao,C. and Polifroni,J., “Multimodal discourse modelling in a multi-user multi-domain environment”, *Proc ICSLP, Philadelphia 1996, pp. 192-195.*
7. Goddeau,D., Meng,H., Polifroni,J., Seneff,S. and Busayapongchai,S., “A form-based dialogue manager for spoken language applications”, *Proc ICSLP, Philadelphia 1996, pp. 701-704.*
8. Schwartz,R., Miller,S., Stallard,D. and Makhoul,J., “Language understanding using hidden understanding models”, *Proc ICSLP, Philadelphia 1996, pp. 997-1000.*
9. Seide,F., Rueber,B. and Kellner,A., “Improving speech understanding by incorporating database constraints and dialogue history”, *Proc. ICSLP, Philadelphia 1996, pp. 1017-1020.*
10. Denecke,M. and Waibel,A., “Dialogue strategies guiding users to their communicative goals”, *Proc. Eurospeech, Rhodes 1997, pp. 1339-1342.*
11. Łukasiewicz, J., “Philosophical remarks on many-valued systems of propositional logic”, 1930, in *McCall,S. (ed), Polish Logic, Oxford University Press, 1967.*
12. Buntschuh,B., Kamm,C., DiFabbrizio,G., Abella,A., Mohri,M., Narayanan,S., Zeljkovic,I., Sharp,R.D., Wright,J.H., Marcus,S., Shaffer,J., Duncan,R. and Wilpon,J.G., “VPQ: a spoken language interface to large scale directory information”, *Proc. ICSLP, Sydney, 1998.*

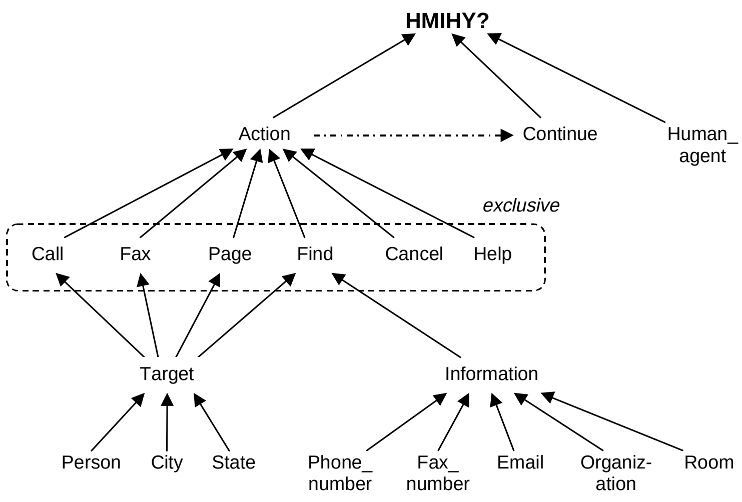

Figure 4: Task structure graph for VPQ.