Considering Multiple Options when Interpreting Spoken Utterances

Sarah George, Ingrid Zukerman, Michael Niemann and Yuval Marom
Faculty of Information Technology
Monash University
Clayton, VICTORIA 3800, AUSTRALIA
{sarahg,ingrid,niemann,yuvalm}@csse.monash.edu.au

Abstract

We describe *Scusi?*, a spoken language interpretation mechanism designed to be part of a robot-mounted dialogue system. *Scusi?*'s interpretation process maps spoken utterances to text, which in turn is parsed and then converted to conceptual graphs. In order to support robust and flexible performance of the dialogue module, *Scusi?* maintains multiple options at each stage of the interpretation process, and uses maximum posterior probability to rank the (partial) interpretations produced at each stage. The time and space requirements of maintaining multiple options are handled by means of an anytime search algorithm. Our evaluation focuses on the impact of the speech recognizer and the search algorithm on *Scusi?*'s performance.

1 Introduction

The DORIS project (*Dialogue Oriented Roaming Interactive System*) aims to develop a spoken dialogue module for a robotic agent. Eventually, this module will be able to engage in a dialogue with users and plan physical actions (by interfacing with a planner). In this paper, we describe *Scusi?*, the speech interpretation module that is being developed within the DORIS framework.

It is widely accepted that spoken dialogue systems are more prone to misinterpretations and partial interpretations than text-based systems. This may be attributed to the state of the art in speech recognition, and to people generally using more informal and less grammatical forms of expression in spoken discourse than in written discourse. In order to handle gracefully the additional uncertainty associated with the interpretation of spoken discourse, a dialogue module should be able to (1) *make decisions on the basis of the state of the interpretation process*, (2) *adjust these decisions dynamically on the basis of new information*, and (3) *recover from flawed or partial interpretations*. For example, if an addressee was reasonably sure that she heard a sentence correctly, her response would differ from the response she would generate if she couldn't quite distinguish between several possible sentences or parts thereof. If new information then came to light (e.g., the speaker just pointed to an object), it could change the certainty of the addressee regarding different interpretations. Still, it is possible that even the preferred interpretation has areas of uncertainty (e.g., it is not clear what the speaker wants done with the object in question). In this case, the addressee can just ask a clarification question regarding the intended action.

Scusi? was designed to enable a dialogue module to achieve the above requirements. *Scusi?*'s interpretation process comprises three main stages (Figure 1): speech recognition, parsing, and semantic interpretation. During semantic interpretation a parse tree is first mapped into a knowledge representation based on *Conceptual Graphs (CGs)* [Sowa, 1984]; this is similar to the assignment of semantic role labels [Gildea and Jurafsky, 2002]. The content of this CG structure is then matched with items and actions in the world (Section 3). Each stage in the interpretation process produces multiple candidate options, which are ranked according to their probability of being intended by the speaker. The probability of a candidate depends on the probability of its parents (generated in the previous stage of the interpretation process) and that of its components (Section 4).

The generation and maintenance of multiple interpretations, and the calculation and update of their probability contribute to the above requirements for a dialogue module as follows.

1. The generation and maintenance of multiple interpretations and the calculation of the probability of an interpretation at each stage of the interpretation process enable the dialogue module to *make decisions on the basis of features of the overall state of the interpretation process*. Examples of such features are: how many highly ranked interpretations there are, how similar they are to each other, and how confident is the system about its interpretations at the different stages. For instance, if there are several top-ranked interpretations, it is reasonable to generate a clarification question that discriminates between them; if all the interpretations produced by the speech recognizer have a low probability, then the dialogue module can initiate a clarification sub-dialogue regarding the spoken utterance; and if the parse tree used for the top interpretation has a low probability, the dialogue module may ask *Scusi?* to perform additional processing using other parse trees.
2. The calculation and update of the probability of interpre-

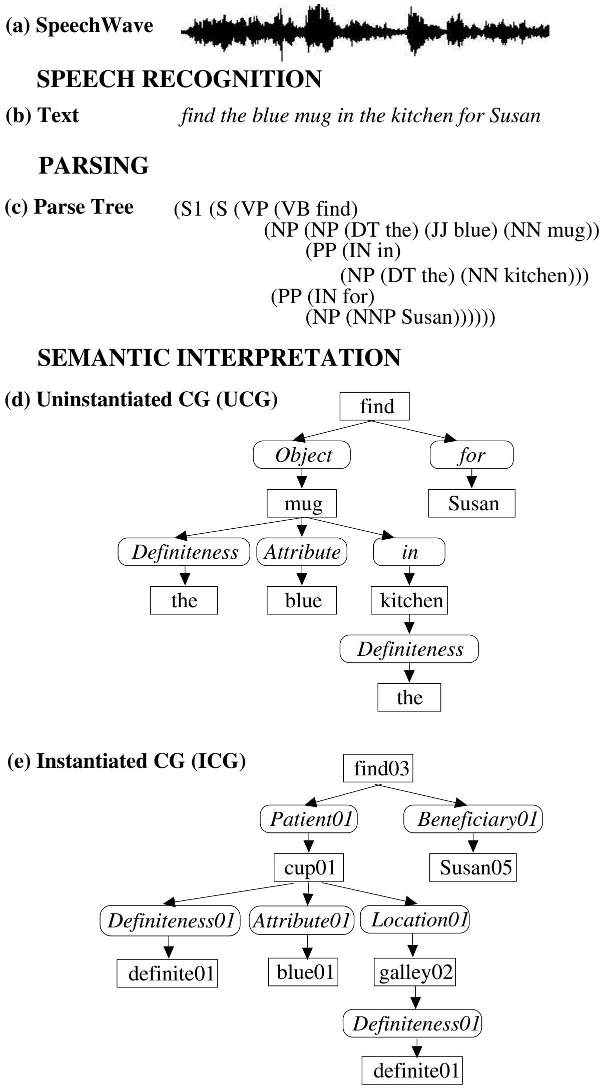


Figure 1: Structures for the interpretation stages

tations supports a dynamic re-ranking of the interpretations as new information becomes available, which in turn enables the dialogue module to *modify its decisions on the fly*. This information may be obtained from additional interpretations generated by *Scusi?* (after it has submitted its current interpretations to the dialogue module), or from new observations, which may be received from a vision module or from a new utterance generated by the user. For example, a clarification question may no longer be required if a newly produced interpretation has a much higher probability than any interpretation generated so far, or if a new visual input can disambiguate between several top-ranked interpretations.

3. As mentioned above, the process of calculating the probability of an interpretation incorporates the calculation of the probability of individual components of the interpretation. This supports the identification of "trusted" (high probability) and "untrusted" (low probability) regions of an interpretation, which enables the dialogue module to select strategies to *recover from flawed or partial interpretations*. For instance, if the speech recognizer is not confident about some words, the resultant interpretation will have low-probability components that correspond to these words; or if a concept in a final interpretation (CG) does not match a domain expectation (e.g., an object to be moved is not movable), the probability of the corresponding component will be low. The identification of these untrusted regions will enable the dialogue module to initiate a focused recovery, such as a clarification question about the components in an untrusted region, e.g., for the first example, it may ask "What do you want me to get?", and for the second example, it may inquire "I understood you want me to move your room. Is this what you meant?".

The rest of this paper is organized as follows. Section 2 presents our interpretation process, followed by a description of conceptual graphs – our knowledge representation formalism. Our probabilistic approach is discussed in Section 4, and an initial evaluation of our interpretation mechanism is presented in Section 5. Section 6 discusses related research, followed by concluding remarks.

2 Multi-Stage Processing

Figure 1 illustrates the stages involved in processing spoken input. The first stage activates an *Automatic Speech Recognizer (ASR)* to generate candidate sequences of words (*Text*) from a *Speech Wave*.[1] Each Text has a score that represents how well its words fit the speech wave. This score is converted into a probability. The word sequences are then parsed using a probabilistic parser, which generates a set of *Parse Trees*.[2]

The last two stages of the interpretation process generate two types of CGs: *Uninstantiated Concept Graphs (UCGs)* and *Instantiated Concept Graphs (ICGs)*. UCGs are obtained from Parse Trees, where one Parse Tree produces one UCG (Section 3.1). UCGs represent mainly syntactic information, i.e., the concepts in a UCG correspond to the words in the parent Parse Tree, and the relations between the concepts are directly derived from syntactic information in the Parse Tree and prepositions. For instance, in the example in Figure 1(c-d), the noun "mug" is mapped to the concept mug, and the preposition "in" in the Parse Tree is mapped to the relation *in* in the UCG. Next, *Scusi?* proposes candidate ICGs for UCGs, where one UCG may yield several ICGs. This is done by nominating *Instantiated Concepts* from DORIS's knowledge base as a potential realization for every *Uninstantiated Concept* in a UCG (Section 3.2). In the example in Figure 1(d-e), the concept mug is mapped to cup01, and the relation *in* in the UCG is mapped to *Location01* in the ICG.

[1] We are currently using ViaVoice (http://www-306.ibm.com/software/voice/viavoice), and trialling Sphinx (http://cmusphinx.sourceforge.net/).

[2] We use Charniak's parser (ftp://ftp.cs.brown.edu/pub/nlparser/) because it can produce partial parses for ungrammatical utterances, and it provides multiple parse trees. This is in line with our approach, which expects multiple options at each stage of the interpretation process.

2.1 Anytime processing

The consideration of all possible options for each stage of the interpretation process is computationally intractable. To address this problem, we have adapted the *anytime* algorithm described in [Niemann *et al.*, 2005], which applies a selection-expansion cycle to build a search graph as follows. The selection step nominates a single sub-interpretation (Speech, Text, Parse Tree or UCG) to expand, and the expansion step generates only one child for that sub-interpretation. The selection step then nominates the next sub-interpretation to expand, which may be the one that was just expanded, its new child, or any other sub-interpretation in the search graph.

The selection-expansion cycle is repeated until one of the following happens: all the options are fully expanded, a time limit is reached, or the system runs out of memory. At that point, the interpretation process returns all the (ranked) interpretations and sub-interpretations obtained so far. This will enable the dialogue module to decide on an action on the basis of the overall state of the interpretation process. If memory hasn't run out, the interpretation process will continue cycling, and if it finds new high-probability interpretations, the dialogue module can adjust its actions accordingly.

Our search algorithm differs from most search algorithms for spoken language interpretation in two respects: (1) it implements a *stochastic optimization strategy*, and (2) it dynamically decides which level in the search graph and which node within this level to expand next.

Stochastic optimization strategies. These strategies, which include simulated annealing and neural nets, occasionally allow low-ranking nodes to generate children. In so doing, these strategies typically avoid getting stuck in local maxima — a problem incurred by greedy algorithms.

Dynamic node selection. Many spoken language interpretation systems apply some type of level-building algorithm [Myers and Rabiner, 1981], which expands each level of the search in turn. In order to curb combinatorial explosion, a beam threshold, which selects the best K options, is used at each level (typically, the value of K is quite small, allowing only the best or top-few interpretations to proceed [Shankaranarayanan and Cyre, 1994; Gorniak and Roy, 2005]). In contrast, our search dynamically determines the stage (level in the search graph) to be expanded, selects a node within that level, and generates one child for this node. In line with our stochastic optimization approach, the first two decisions are probabilistic, choosing preferred options most often, but not always. In order to encourage the early generation of complete interpretations, preference is given to later stages in the search (e.g., expanding UCGs rather than Texts). Within a level, nodes with a proven "track record" are preferred, i.e., nodes that have previously produced high-probability children. This heuristic cannot be used by a level-building algorithm, as information about later stages is not available to earlier stages. In Section 5, we compare the performance of our search with that of a level-building algorithm.

3 Conceptual Graphs

Conceptual graphs represent entities and the relationships between them.[3] For instance, the CG in Figure 1(e) indicates that there are two concepts find03 and cup01 that have a *Patient01* relationship. Every relationship in a CG must have at least one parent concept and one child concept, e.g., the *Patient01* relationship in Figure 1(e) has concept find03 as a parent and cup01 as a child. However, a concept can exist in isolation without any relationships. This supports phrases as well as single-word utterances like "yes", "there" and "Mary".

3.1 Uninstantiated Conceptual Graphs

A UCG represents concepts and relationships that can be obtained directly from the Parse Tree (without resorting to domain knowledge). Most phrases in a Parse Tree map to a concept node representing their head-word. If a phrase governs a word or phrase other than its head-word, then the phrase's concept node is joined to the other word or phrase's concept node via a relationship node. For instance, the adjective (JJ) *"blue"* in Figure 1(c), which is governed by the same NP as the noun (NN) *"mug"*, is connected to the mug concept in the UCG in Figure 1(d) by means of an *Attribute* relationship node (*Attribute* is the default relationship). Linguistic details such as part-of-speech and phrasal category are retained as features of the concepts. Prepositions are treated as defining a relationship node between two concepts, rather than mapping to concept nodes, e.g., *for* represents the relationship between find and Susan in Figure 1(d). It is worth noting that slightly different Parse Trees may yield the same UCG. For instance, the blue concept node in Figure 1(d) could also be generated from an Adjectival Phrase, instead of a stand-alone adjective (JJ) adjunct to the NP.

This representation allows *Scusi?* to accept and combine information from different types of sentences and input modalities. For spoken input, our mapping from Parse Tree to UCG handles declarative, imperative and interrogative sentences, as well as single words. In the future, *Scusi?* is expected to interact with the scene analysis component of a robot's vision system. This component will return objects and relationships such as *Coordinates*, *Colour* or *Shape*, which can readily map to UCGs.

3.2 Instantiated Conceptual Graphs and the Knowledge Base

The generation of an ICG requires the selection of an *Instantiated Concept* from the knowledge base for each *Uninstantiated Concept* in a UCG. The knowledge base contains entries for the following types of concepts.

- Specific real-world objects, e.g., cup03, Susan05;
- general objects that have default features, e.g., CupClass01, which has features like container=Y, movable=Y, shape=cylinder and size=small;

[3] Our knowledge representation is structurally like CGs, but the relations are inspired by the Verb Semantic Classes from EAGLES96 (http://www.ilc.cnr.it/EAGLES96/rep2/node10.html) and by FrameNet categories (http://framenet.icsi.berkeley.edu/).

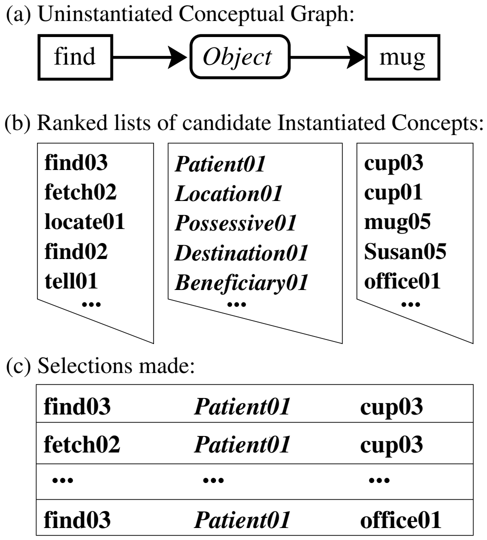


Figure 2: Selection of Instantiated Concepts

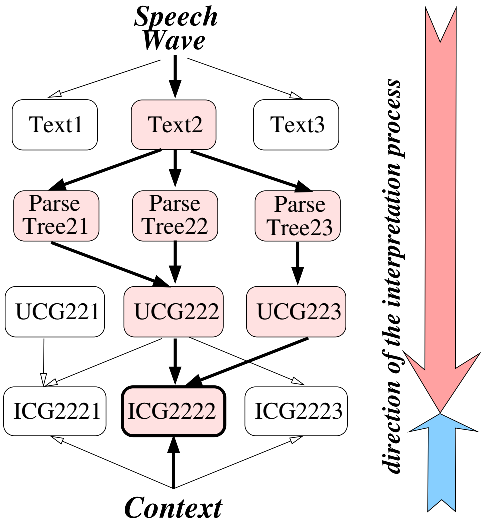


Figure 3: Interpretation process

- abstract attributes like blue01 and quickly01;
- actions known to the system, e.g., find01 for locating a place and reporting its whereabouts, as in “find an office for Susan”, and find03 for retrieving an object, as in “find a cup for Susan”; and
- instantiated Relationships, e.g., roles like *Patient01*, *Destination01* and *Beneficiary01*.

The process for postulating Instantiated Concepts for Uninstantiated ones is similar to that used for suppositions in [George *et al.*, 2005]. Each Uninstantiated Concept in a UCG is associated with a list of Instantiated Concepts (Figure 2(b)). Each entry in the list is assigned a probability on the basis of how well it matches the Uninstantiated Concept (Section 4). To generate an ICG, one Instantiated Concept is selected from the list of each Uninstantiated Concept in the parent UCG, starting with the higher-ranked combinations (Figure 2(c)). Subject to time and memory limitations, all combinations of Instantiated Concepts may eventually be considered.

4 Probabilities of Interpretations

Scusi? ranks candidate ICGs according to their posterior probability in light of a given Speech Wave and conversational context. At present, the context is obtained from concept and relation instances in the system's knowledge base, which in the absence of other information are equiprobable. We are currently in the process of incorporating salience from dialogue history into our formalism, such that it influences the prior probability of mentioning a concept or relation. In the future, we will also include information from the robot's vision system.

As seen in Section 2, the interpretation process goes mainly from evidence (Speech Wave) to ICG (thick arrows on the right-hand-side of Figure 3). The ASR provides probabilities from Speech Wave to Text (its scores are directly translated to probabilities), and the probabilistic parser from Text to Parse Tree. We therefore calculate the posterior probability of an ICG as follows.

$$\Pr(\mathit{ICG}|\mathit{Speech}) \cong \alpha \times \sum_{\mathit{txt},\mathit{prsTr},\mathit{ucg}} \left\{ \begin{array}{l} \Pr(\mathit{ICG}|\mathit{UCG},\mathit{Context}) \times \Pr(\mathit{UCG}|\mathit{ParseTr}) \times \\ \Pr(\mathit{ParseTr}|\mathit{Text}) \times \Pr(\mathit{Text}|\mathit{Speech}) \end{array} \right\} \tag{1}$$

where α is a normalizing constant.

The summation is required since, as seen in Figure 3, a sub-interpretation may have multiple parents. The retention of all sub-interpretations regardless of their probability enables us to get a true measure of the overall probability of any interpretation with multiple paths to the Speech Wave.

Further, since a UCG is generated algorithmically from a Parse Tree, $\Pr(\mathit{UCG}|\mathit{ParseTr}) = 1$. Thus, the only outstanding issue is the calculation of $\Pr(\mathit{ICG}|\mathit{UCG},\mathit{Context})$. To perform this calculation we make the following simplifying assumptions.

- $\Pr(\mathit{ICG}|\mathit{UCG},\mathit{Context})$ can be calculated separately for each node (concept or relation) in the ICG; and
- given a source UCG and the Context, the probability of each node N_i^{ICG} in an ICG depends on its corresponding node in the UCG (N_i^{UCG}), its neighbouring nodes in the ICG ($\mathit{nbours}(N_i^{ICG})$), and its prior probability in the Context.

These assumptions yield the following formulation.

$$\Pr(\mathit{ICG}|\mathit{UCG},\mathit{Context}) = \prod_{i=1}^{n} \Pr(N_i^{ICG}|N_i^{UCG}, \mathit{nbours}(N_i^{ICG}), \mathit{Context}) \tag{2}$$

where n is the number of nodes in the ICG.

By making some conditional independence assumptions, we obtain the following approximation for this equation.

$$\Pr(\mathit{ICG}|\mathit{UCG},\mathit{Context}) = \beta \times \prod_{i=1}^{n} \left\{ \begin{array}{l} \Pr(N_i^{UCG}|N_i^{ICG}) \times \Pr(\mathit{nbours}(N_i^{ICG})|N_i^{ICG}) \times \\ \Pr(N_i^{ICG}|\mathit{Context}) \end{array} \right\} \tag{3}$$

Table 1: Features for sample concepts in the UCG and ICG

Stage	called	PoS	rel	cg-role
UCG	“find”	{S1,S,VP,VB,word}	–	concept
UCG	“fine”	{S1,ADJP,JJ,word}	–	concept
ICG find01	“find, locate”	{VP,VB,VBZ,VBP, VBN,VBG,VBD}	–	concept
ICG find03	“find, locate”	{VP,VB,VBZ,VBP, VBN,VBG,VBD}	–	concept

where β is a normalizing constant.

- The third factor in this product contains the prior probability of node i in ICG, which reflects the salience of the concept or relation in question in the current context.
- The second factor reflects how reasonable it is to put the concepts of the ICG together, i.e., it encodes the extent to which each node in the ICG matches the requirements of other nodes. For example, find03 in Figure 1(e) expects a *Patient01* relationship and a *Beneficiary01* relationship with other concepts.
- The first factor in the product represents how well a candidate ICG node matches a source UCG node. We have found it useful to consider four features of these nodes to determine the goodness of this match (described below). The values of these features for a UCG node are obtained from the parser, and the possible values that can be taken by a candidate ICG node are stored in the knowledge base.

1. called – the lexical items associated with a concept, e.g., “find” and “mug” in the UCG in Figure 1(d). This feature is used to determine whether the words in a user’s utterance could be used to designate a candidate concept or relation in the knowledge base. In the future, we intend to complement this feature with similarity metrics such as those discussed in [Pedersen *et al.*, 2004].
2. PoS – part of speech. This feature is more forgiving than called, because only some of the PoS-tags returned by a parser inform the matching process.
3. relation – syntactic relation, e.g., *Object* and *Attribute* in the UCG in Figure 1(d). This feature has a value when the parser provides information about the type of a relation between concepts. Like called, this feature is used to determine whether the relations in a user’s utterance could be used to designate a candidate relation in the knowledge base.
4. cg-role – the semantic role of a node, i.e., concept or relation. This feature is crucial for determining whether a candidate ICG node could possibly match an uttered word.

To illustrate the calculation of these factors, consider a situation where given the input in Figure 1, the ASR has alternatively heard “find” and “fine” as the first word. Table 1 shows the above features for “find” and “fine” in the UCG, and for two candidate domain actions in the ICG: find01 and find03.

Both UCG words match the cg-role and relation features for both domain actions. However, “fine” does not match the called and PoS features (the parse that produced the UCG in this example does not have “fine” as a verb). Hence, the UCG with “find” has a higher probability of being the parent of an ICG that has find01 as a candidate action, and an ICG that has find03 as a candidate action. After calculating the first factor in Equation 3, the find nodes in these ICGs are equiprobable. However, the second factor discriminates between these domain actions. This is done through the expectations of action-relation pairs. For example, find03 expects an inanimate object as a patient, while find01 expects a place. Since “cup01” is an object, find03 has a higher probability. At present, *Scusi?* interprets utterances in isolation, hence dialogue context has no influence.

5 Evaluation

To evaluate *Scusi?*’s speech interpretation performance, we used 27 utterances, which were based on the TRAINS92 corpus [Allen *et al.*, 1996] and were spoken by one of the authors. These utterances were selected due to their simplicity, so that they are easy to parse. The utterances had different lengths, ranging from 3 to 13 words, e.g., *“back to Illinois”* and *“bring the boxcar back to Avon to fill the boxcar up with bananas”*. The knowledge base had 139 concepts (e.g., go0, town_Avon).

Our evaluation focuses on *Scusi?*’s ability to generate the intended interpretation (hence measures of partial matches, such as Word Error Rate, are not appropriate). We defined two gold standards as follows. Each utterance had one Gold Text which was the original TRAINS text (the ASR could produce the Gold Text for 23 of the 27 utterances considered — our evaluation is based on those 23 utterances). In addition, each utterance had zero or more Gold ICGs among the ICGs generated by *Scusi?* (sometimes *Scusi?* could not find a correct ICG, and sometimes there were several appropriate interpretations). The correctness of an ICG was determined on the basis of the knowledge base. If an Uninstantiated Concept did not have a corresponding concept in the knowledge base, then the Gold ICG mapped it to a generic concept, e.g., unknown_concept, unknown_noun. Otherwise, the Gold ICGs contain concepts from the knowledge base that are valid representations of what the speaker uttered, e.g., the Gold ICG for “go to Corning” is go0 $\rightarrow$Destination_Relationship $\rightarrow$town_Corning.

Our evaluation focuses on two aspects of *Scusi?*’s operation: (1) the effect of speech recognition performance on interpretation performance, and (2) our search algorithm.

Effect of speech recognition performance. Our ASR often produces a large number of options for a spoken utterance. For instance, the utterance *“We were going to take the red engine”* yields 7,682 options, and *“Pick up a tanker in Corning I guess”* produces 53,762 alternatives. Figure 4 shows the number of Gold ICGs found by *Scusi?* as a function of the error percentage of the ASR. This is the proportion of incorrect outputs produced by the ASR that were used in the interpretation process. 0% error means that the ASR was deemed to return only the Gold Text, and 100% means that the ASR Gold

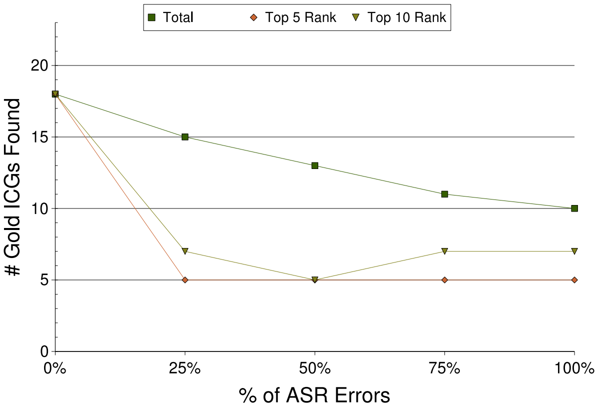


Figure 4: Effect of speech recognition errors, 300 iterations

Text was mixed in among all the erroneous options produced by the ASR (clearly, when thousands of options are produced by the ASR, not all of them can be considered in the available time). The 25, 50 and 75 error percentages were produced by selecting every fourth, second, and three out of four erroneous options produced by the ASR. *Scusi?* was run for 300 iterations, which is our current default setting for tests.

The graph also shows how error percentage affects the number of Gold ICGs with *average rank* ≤ 5 and ≤ 10, where average rank is the average of the ranks of equiprobable interpretations, e.g., if there are three equiprobable interpretations ranked 1, 2 and 3, their average rank is 2. We used *average rank* rather than raw rank because it is often the case that clusters of interpretations have the same probability.

As expected, ASR accuracy has a significant influence on interpretation performance, with accurate recognition yielding 18 Gold ICGs out of the possible 23, and all these ICGs being ranked top 5. As ASR accuracy drops, so does interpretation performance. However, this deterioration is graceful in terms of our system's ability to find Gold ICGs, while it is sudden for average rank. Further, the number of Gold ICGs with ranks ≤ 5 and ≤ 10 remains constant between ASR error of 25% and 100%. We propose to investigate two ways to obtain better output from the ASR: we are considering procedures for filtering the options returned by ViaVoice, and in parallel we are experimenting with a different ASR.

***Scusi?*'s search algorithm.** Figure 5 compares the performance of our search algorithm with that of a level-building algorithm that uses beams of different sizes. Since the level-building algorithm expands each level in turn, a sub-interpretation does not have information about the performance of its children. Hence, unlike *Scusi?*-search, beam search selects the top sub-interpretations to be expanded on the basis on their probability only (Section 2). Figure 5(a) shows the number of ASR Gold Texts found by both algorithms (from the possible 23), and Figure 5(b) shows the number of Gold ICGs. Both figures show the total number of Golds found, and the number with average ranks of 1, ≤ 5 and ≤ 10. The plain-coloured bars show the performance of the level-building algorithm for beams of size 1, 5 and 10, and the bars with diagonal stripes show the performance of the "corresponding" *Scusi?*-search. This corresponding search was defined in order to make the comparison fair — the number of iterations it performs is equal to the number of iterations performed by the beam search. For example, Beam-1 means that only the top-ranked option was expanded by the beam search at every stage, which is equivalent to 6 *Scusi?* iterations; Beam-10 is equivalent to *Scusi?*-350 (note that *Scusi?*-350 finds 12 Gold ICGs, compared to 10 Gold ICGs found by *Scusi?*-300, plotted in Figure 4).

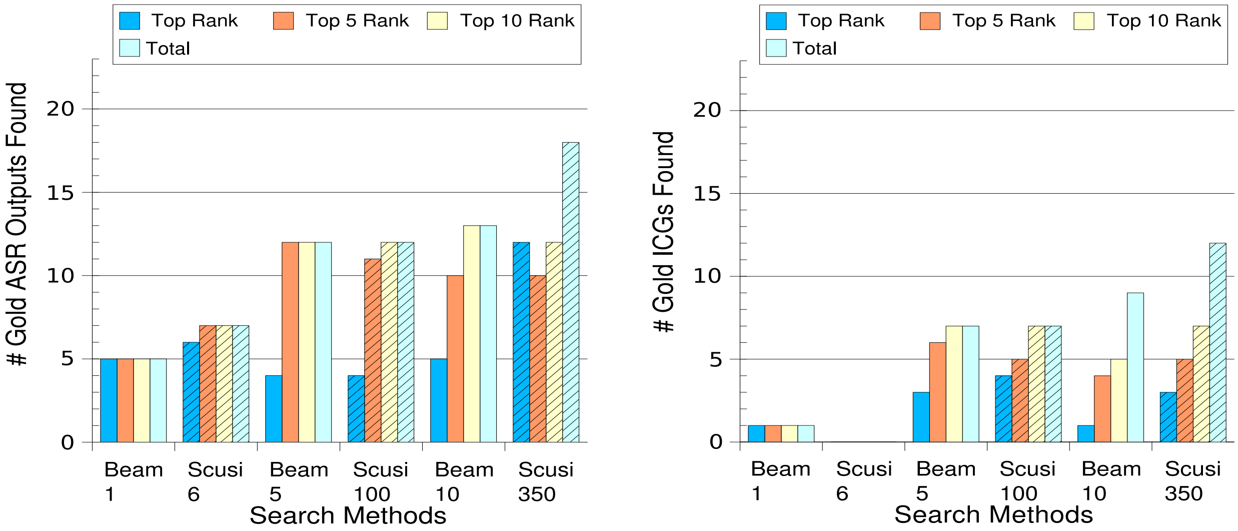
As seen in Figure 5, the performance of *Scusi?*-search is slightly better than that of beam search, in particular for *Scusi?*-350 versus Beam-10. *Scusi?*-350 found 18 Gold Texts, which led to 12 Gold ICGs, while Beam-10 found 13 Gold Texts, which led to 9 Gold ICGs. Note that the additional Gold Texts found by *Scusi?* have ranks greater than 10, and hence are unlikely to be found by a rigid beam search. This indicates that *Scusi?*'s flexible expansion procedure is a promising approach. Additionally, *Scusi?*'s anytime performance (Section 2) makes it more responsive to its operating conditions than systems governed by arbitrary thresholds (e.g., beam size). Hence, we consider this approach worth pursuing. Also note that our results, both for beam search and *Scusi?*-search, are heavily influenced by our calculations of the probability of a sub-interpretation. We expect that additional information brought to bear to these calculations, such as corpus-based statistics, will yield improvements in interpretation performance.

6 Related Research

This research extends the work described in [Niemann *et al.*, 2005] in its use of CGs as its main knowledge representation formalism. CGs were chosen, instead of the simple frames used by Niemann *et al.*, due to their higher expressive power (the relationship between CGs and predicate calculus is discussed in [Dau, 2001]). The use of CGs in turn affects the calculation of the posterior probability of an interpretation.

Miller *et al.* [1996] and He and Young [2003] also applied a probabilistic approach for the interpretation of utterances from the ATIS corpus, and Pfleger *et al.* [2003] used this approach to interpret multi-modal input (but using a scoring function, rather than probabilities). However, these three projects use semantic grammars for parsing, while *Scusi?*'s interpretation process initially uses generic, syntactic tools, and incorporates semantic- and domain-related information only in the final stage of the process. Knight *et al.* [2001] compared the performance of a grammar-based dialogue system to that of a system based on a statistical language model and a robust phrase-spotting grammar. The latter performed better for relatively unconstrained utterances by users unfamiliar with the system. Our probabilistic approach and intended audience are in line with this finding.

Like us, Fischer *et al.* [1998] regarded speech interpretation as an optimization task. They achieved anytime performance by employing a stochastic optimization method which considers multiple interpretations and expands "sub-optimal" candidates. However, their use of statistical information is fundamentally different from ours, as they use the results of



(a) Number of ASR Gold Texts

(b) Number of Gold ICGs

Figure 5: Comparison between *Scusi?*-search and beam search, 100% ASR error

statistical analysis to prime the interpretation process. Additionally, they worked on railway schedule queries, which are stylistically constrained.

Sowa and Way [1986] and Shankaranarayanan and Cyre [1994] used conceptual graphs for discourse interpretation. Both used a predefined set of canonical graphs to define the semantics of the base concepts in their system. *Scusi?* differs from both of these systems in its use of the UCG as an intermediate stage that is independent from the semantic- and domain-knowledge in the knowledge base. From a processing point of view, Shankaranarayanan and Cyre considered only the first parse tree that supports an acceptable interpretation, rather than retaining multiple parse trees. Sowa and Way allowed multiple interpretations, but applied a filtering mechanism that removed parses that failed semantic expectations. *Scusi?* does not apply such filtering, allowing possibly flawed candidates to undergo a deeper examination.

Our work resembles that of Horvitz and Paek [1999] and Gorniak and Roy [2005] in its integration of context-based expectations with alternatives obtained from spoken utterances. Gorniak and Roy use a probabilistic parser like ours, but they restrict the search space by training the parser on a corpus of human interactions relating to a computer game, and provide tightly constrained domain expectations based on the appropriate actions at particular stages in the game. In addition, they allow only the most probable parse state to generate an interpretation. In contrast, we do not restrict our expected input, we only factor in domain knowledge in the final stage of the interpretation, and dis-preferred sub-interpretations are allowed to proceed to the next stage. The differences between these approaches highlight important trade-offs between processing speed, flexibility and robustness.

Horvitz and Paek focused on higher level informational goals than those addressed in this paper, using a single output produced by a parser as linguistic evidence for their goal recognition system. An important aspect of their work, which we hope to incorporate into our dialogue module in the future, is their use of a utility-based decision procedure to determine the system's actions on the basis of the probabilities of interpretations.

7 Conclusion

We have presented a multi-stage interpretation process that maintains multiple options at each stage of the process, and uses maximum posterior probability to rank the (partial) interpretations produced at each stage. We have argued that these features support the following desirable behaviours in a dialogue module: making decisions on the basis of the state of the interpretation process, adjusting these decisions dynamically on the basis of new information, and recovering from flawed or partial interpretations.

The time and space requirements of maintaining multiple options are handled by means of an anytime search algorithm. Our algorithm dynamically decides which level to expand in a search graph, and which node within a level. This supports flexible behaviour that takes into account a system's operating constraints. Additionally, our algorithm employs a stochastic optimization method, which allows the examination of sub-optimal sub-interpretations.

Our evaluation considered two aspects of the interpretation of spoken discourse: (1) impact of ASR performance, and (2) search algorithm. As expected, ASR performance affects our system's interpretation performance overall. However, in our experiments, our system's ability to produce highly-ranked interpretations was invariant for ASR error percentages between 25% and 100%. Our search algorithm performed slightly better than a traditional beam-search approach. This, together with our algorithm's flexibility, indicate that our approach is worth pursuing.

Acknowledgments

This research was supported in part by the ARC Centre for Perceptive and Intelligent Machines in Complex Environments. The authors thank Eugene Charniak for his modifications to his probabilistic parser, Charles Prosser for his assistance in extracting multiple texts from ViaVoice, David Albrecht for his advice on the probabilistic formulation, and James Allen for permission to use and modify the transcripts from the TRAINS project [Allen *et al.*, 1996].

References

[Allen *et al.*, 1996] J.F. Allen, B.W. Miller, E.K. Ringger, and T. Sikorski. A robust system for natural spoken dialogue. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 62–70, Santa Cruz, California, 1996.

[Dau, 2001] F. Dau. Concept graphs and predicate logic. In *ICCS 2001 – Proceedings of the 9th International Conference on Conceptual Structures*, Stanford, California, 2001.

[Fischer *et al.*, 1998] J. Fischer, J. Haas, E. Nöth, H. Niemann, and F. Deinzer. Empowering knowledge based speech understanding through statistics. In *ICSLP'98 – Proceedings of the Fifth International Conference on Spoken Language Processing*, volume 5, pages 2231–2235, Sydney, Australia, 1998.

[George *et al.*, 2005] S. George, I. Zukerman, and M. Niemann. Modeling suppositions in users' arguments. In *UM05 – Proceedings of the 10th International Conference on User Modeling*, pages 19–29, Edinburgh, Scotland, 2005.

[Gildea and Jurafsky, 2002] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

[Gorniak and Roy, 2005] P. Gorniak and D. Roy. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *ICMI'05 – Proceedings of the Seventh International Conference on Multimodal Interfaces*, Trento, Italy, 2005.

[He and Young, 2003] Y. He and S. Young. A data-driven spoken language understanding system. In *ASRU'03 – Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, St. Thomas, US Virgin Islands, 2003.

[Horvitz and Paek, 1999] E. Horvitz and T. Paek. A computational architecture for conversation. In *UM99 – Proceedings of the Seventh International Conference on User Modeling*, pages 201–210, Banff, Canada, 1999.

[Knight *et al.*, 2001] S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. Comparing grammar-based and robust approaches to speech understanding: A case study. In *Proceedings of Eurospeech 2001*, Aalborg, Denmark, 2001.

[Miller *et al.*, 1996] S. Miller, D. Stallard, R. Bobrow, and R. Schwartz. A fully statistical approach to natural language interfaces. In *ACL96 – Proceedings of the 34th Conference of the Association for Computational Linguistics*, pages 55–61, Santa Cruz, California, 1996.

[Myers and Rabiner, 1981] C. Myers and L. Rabiner. A level building dynamic time warping algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2:284–297, 1981.

[Niemann *et al.*, 2005] M. Niemann, S. George, and I. Zukerman. Towards a probabilistic, multi-layered spoken language interpretation system. In *Proceedings of the Fourth IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 8–15, Edinburgh, Scotland, 2005.

[Pedersen *et al.*, 2004] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity – measuring the relatedness of concepts. In *AAAI-04 – Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 25–29, San Jose, California, 2004.

[Pfleger *et al.*, 2003] N. Pfleger, R. Engel, and J. Alexandersson. Robust multimodal discourse processing. In *Proceedings of the Seventh Workshop on the Semantics and Pragmatics of Dialogue*, Saarbrücken, Germany, 2003.

[Shankaranarayanan and Cyre, 1994] S. Shankaranarayanan and W.R. Cyre. Identification of coreferences with conceptual graphs. In *ICCS'94 – Proceedings of the Second International Conference on Conceptual Structures*, College Park, Maryland, 1994.

[Sowa and Way, 1986] J.F. Sowa and E.C. Way. Implementing a semantic interpreter using conceptual graphs. *IBM Journal of Research and Development*, 30(1):57–69, 1986.

[Sowa, 1984] J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.