# Towards the Interpretation of Utterance Sequences in a Dialogue System

**Ingrid Zukerman and Patrick Ye and Kapil Kumar Gupta and Enes Makalic**
Faculty of Information Technology
Monash University
Clayton, VICTORIA 3800, Australia
ingrid@infotech.monash.edu.au, {ye.patrick,kapil.k.gupta,emakalic}@gmail.com

## Abstract

This paper describes a probabilistic mechanism for the interpretation of sentence sequences developed for a spoken dialogue system mounted on a robotic agent. The mechanism receives as input a sequence of sentences, and produces an interpretation which integrates the interpretations of individual sentences. For our evaluation, we collected a corpus of hypothetical requests to a robot. Our mechanism exhibits good performance for sentence pairs, but requires further improvements for sentence sequences.

## 1 Introduction

*DORIS* (Dialogue Oriented Roaming Interactive System) is a spoken dialogue system under development, which will eventually be mounted on a household robot. The focus of our current work is on *DORIS*'s language interpretation module called *Scusi?*. In this paper, we consider the interpretation of a sequence of sentences.

People often utter several separate sentences to convey their wishes, rather than producing a single sentence that contains all the relevant information (Zweig et al., 2008). For instance, people are likely to say "Go to my office. Get my mug. It is on the table.", instead of "Get my mug on the table in my office". This observation, which was validated in our corpus study (Section 4), motivates the mechanism for the interpretation of a sequence of sentences presented in this paper. Our mechanism extends our probabilistic process for interpreting single spoken utterances (Zukerman et al., 2008) in that (1) it determines which sentences in a sequence are related, and if so, combines them into an integrated interpretation; and (2) it provides a formulation for estimating the probability of an interpretation of a sentence sequence, which supports the selection of the most probable interpretation. Our evaluation demonstrates that our mechanism performs well in understanding textual sentence pairs of different length and level of complexity, and highlights particular aspects of our algorithms that require further improvements (Section 4).

In the next section, we describe our mechanism for interpreting a sentence sequence. In Section 3, we present our formalism for assessing the probability of an interpretation. The performance of our system is evaluated in Section 4, followed by related research and concluding remarks.

## 2 Interpreting a Sequence of Utterances

*Scusi?* employs an anytime algorithm to interpret a sequence of sentences (**Algorithm 1**). The algorithm generates interpretations until time runs out (in our case, until a certain number of iterations has been executed). In Steps 1–5, Algorithm 1 processes each sentence separately according to the interpretation process for single sentences described in (Zukerman et al., 2008).[1] Charniak's probabilistic parser[2] is applied to generate parse trees for each sentence in the sequence. The parser produces up to $N$ ($= 50$) parse trees for each sentence, associating each parse tree with a probability. The parse trees for each sentence are then iteratively considered in descending order of probability, and algorithmically mapped into *Uninstantiated Concept Graphs (UCGs)* — a representa-

[1] Although *DORIS* is a spoken dialogue system, our current results pertain to textual input only. Hence, we omit the aspects of our work pertaining to spoken input.

[2] ftp://ftp.cs.brown.edu/pub/nlparser/

*Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue*, pages 46–53,
Queen Mary University of London, September 2009. ©2009 Association for Computational Linguistics

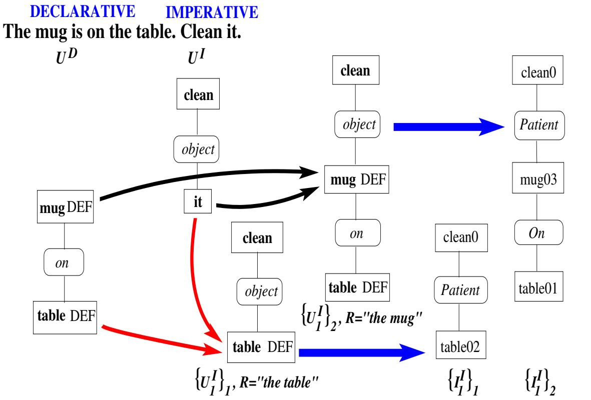
**Algorithm 1** Interpret a sentence sequence

```
Require: Sentences $T_1, \ldots, T_n$
  { Interpret Sentences }
 1: for all sentences $T_i$ do
 2:     Generate parse trees $\{P_i\}$, and UCGs $\{U_i\}$
 3:     Generate candidate modes $\{M_i\}$
 4:     For each identifier $j$ in $T_i$, generate candidate referents $\{R_{ij}\}$
 5: end for
  { Combine UCGs }
 6: while there is time do
 7:     Get $\{(U_1, M_1, R_1), \ldots, (U_n, M_n, R_n)\}$ — a sequence of tuples (one tuple per sentence)
 8:     Generate $\{U^D\}$, a sequence of declarative UCGs, by merging the declarative UCGs in $\{(U_i, M_i, R_i)\}$ as specified by their identifier-referent pairs and modes
 9:     Generate $\{U^I\}$, a sequence of imperative UCGs, by merging each imperative UCG in $\{(U_i, M_i, R_i)\}$ with declarative UCGs as specified by their identifier-referent pairs and modes
10:     Generate candidate ICG sequences $\{I^I_j\}$ for the sequence $\{U^I\}$
11:     Select the best sequence of ICGs $\{I^{I*}\}$
12: end while
```

tion based on Concept Graphs (Sowa, 1984) — one parse tree yielding one UCG (but several parse trees may produce the same UCG). UCGs represent syntactic information, where the concepts correspond to the words in the parent parse tree, and the relations are derived from syntactic information in the parse tree and prepositions (Figure 1(a) illustrates UCGs $U^D$ and $U^I$ generated from the sentences “The mug is on the table. Clean it.”).

Our algorithm requires sentence mode (declarative, imperative or interrogative[3]), and resolved references to determine how to combine the sentences in a sequence. Sentence mode is obtained using a classifier trained on part of our corpus (Section 2.2). The probability distribution for the referents of each identifier is obtained from the corpus and from rules derived from (Lappin and Leass, 1994; Ng et al., 2005) (Section 2.3).

At this point, for each sentence $T_i$ in a sequence, we have a list of UCGs, a list of modes, and lists



(a) Declarative and imperative UCGs (b) Merged UCGs (c) Candidate ICGs

Figure 1: Combining two sentences

of referents (one list for each identifier in the sentence). In Step 7, Algorithm 1 generates a tuple $(U_i, M_i, R_i)$ for each sentence $T_i$ by selecting from these lists a UCG, a mode and a referent for each identifier (yielding a list of identifier-referent pairs). Each element in each $(U, M, R)$ tuple is iteratively selected by traversing the appropriate list in descending order of probability. For instance, given sentences $T_1, T_2, T_3$, the top UCG for $T_1$ is picked first, together with the top mode and the top identifier-referent pairs for that sentence (likewise for $T_2$ and $T_3$); next the second-top UCG is chosen for $T_1$, but the other elements remain the same; and so on.

Once the $(U, M, R)$ tuples have been determined, the UCGs for the declarative sentences are merged in the order they were given (Step 8). This is done by first merging a pair of declarative UCGs, then merging the resultant UCG with the next declarative UCG, and so on. The idea is that if the declarative sentences have co-referents, then the information about these co-referents can be combined into one representation. For example, consider the sequence “The mug is on the table. It is blue. Find it. The mug is near the phone. Bring it to me.” Some of the UCG sequences obtained from the declarative sentences (first, second and fourth) are:

$\{U^D_1\}_1$={mug(*CLR* blue)-
(*on*-table & *near*-phone)}

$\{U^D_1\}_2$={mug-(*on*-table(*CLR* blue) &
*near*-phone)}

$\{U^D_1, U^D_2\}_3$={mug(*CLR* blue)-*on*-table,
mug-*near*-phone}.[4]

[3] Interrogatives are treated as imperatives at present, so in the remainder of the paper we do not mention interrogatives.

[4] The different notations are because colour (and size) are properties of objects, while prepositions indicate relations.

The first two sequences contain one declarative merged UCG, and the third contains two UCGs.

In Step 9, Algorithm 1 considers a UCG for each imperative sentence in turn, and merges it with declarative UCGs (which may have resulted from a merger), as specified by the modes and identifier-referent pairs of the sentences in question. For example, consider the sentence sequence "Find my mug. It is in my office. Bring it." One of the $(U, M, R)$-tuple sequences for this instruction set is
$\{$(find-*obj*-mug-*owner*-me, imperative, NIL),
(it1-*in*-office-*owner*-me, declarative, it1-mug),
(bring-*obj*-it2, imperative, it2-mug)$\}$.

After merging the first two UCGs (imperative-declarative), and then the second and third UCGs (declarative-imperative), we obtain the imperative UCG sequence $\{U_1^I, U_2^I\}$:
$U_1^I$=find-*obj*-mug-(*owner*-me &
*in*-office-*owner*-me)
$U_2^I$=bring-*obj*-mug-(*in*-office-*owner*-me).

This process enables *Scusi?* to iteratively merge ever-expanding UCGs with subsequent UCGs, eventually yielding UCG sequences which contain detailed UCGs that specify an action or object. A limitation of this merging process is that the information about the objects specified in an imperative UCG is not aggregated with the information about these objects in other imperative UCGs, and this sometimes can cause the merged imperative UCGs to be under-specified. This limitation will be addressed in the immediate future.

After a sequence of imperative UCGs has been generated, candidate *Instantiated Concept Graphs (ICGs)* are proposed for each imperative UCG, and the most probable ICG sequence is selected (Steps 10–11 of Algorithm 1). We focus on imperative UCGs because they contain the actions that the robot is required to perform; these actions incorporate relevant information from declarative UCGs. ICGs are generated by nominating different instantiated concepts and relations from the system's knowledge base as potential realizations for each concept and relation in a UCG (Zukerman et al., 2008); each UCG can generate many ICGs. Since this paper focuses on the generation of UCG sequences, the generation of ICGs will not be discussed further.

### 2.1 Merging UCGs

Given tuples $(U_i, M_i, R_i)$ and $(U_j, M_j, R_j)$ where $j > i$, pronouns and one-anaphora in $U_j$ are replaced with their referent in $U_i$ on the basis of the set of identifier-referent pairs in $R_j$ (if there is no referent in $U_i$ for an identifier in $U_j$, the identifier is left untouched). $U_i$ and $U_j$ are then merged into a UCG $U_m$ by first finding a node $n$ that is common to $U_i$ and $U_j$, and then copying the sub-tree of $U_j$ whose root is $n$ into a copy of $U_i$. If more than one node can be merged, the node (head noun) that is highest in the $U_j$ structure is used. If one UCG is declarative and the other imperative, we swap them if necessary, so that $U_i$ is imperative and $U_j$ declarative.

For instance, given the sentences "The mug is on the table. Clean it." in Figure 1, Step 4 of Algorithm 1 produces the identifier-referent pairs $\{$(it, mug), (it, table)$\}$, yielding two intermediate UCGs for the imperative sentence: (1) clean-*object*-mug, and (2) clean-*object*-table. The first UCG is merged with a UCG for the declarative sentence using `mug` as root node, and the second UCG is merged using `table` as root node. This results in merged UCG sequences (of length 1) corresponding to "Clean the table" and "Clean the mug on the table" ($\{U_1^I\}_1$ and $\{U_1^I\}_2$ respectively in Figure 1(b), which in turn produce ICG sequences $\{I_1^I\}_1$ and $\{I_1^I\}_2$ in Figure 1(c), among others).

### 2.2 Determining modes

We use the MaxEnt classifier[5] to determine the mode of a sentence. The input features to the classifier (obtained from the highest probability parse tree for this sentence) are: (1) top parse-tree node; (2) position and type of the top level phrases under the top parse-tree node, e.g., (0, NP), (1, VP), (2, PP); (3) top phrases under the top parse-tree node reduced to a regular expression, e.g., VP-NP$^+$ to represent, say, VP NP NP; (4) top VP head – the head word of the first top level VP; (5) top NP head – the head word of the first top level NP; (6) first three tokens in the sentence; and (7) last token in the sentence. Using leave-one-out cross validation, this classifier has an accuracy of 97.8$\%$ on the test data — a 30$\%$ improvement over the majority class (imperative) baseline.

### 2.3 Resolving references

*Scusi?* handles pronouns, one-anaphora and NP identifiers (e.g., "the book"). At present, we consider only precise matches between NP identifiers

[5] `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`

and referents, e.g., “the cup” does not match “the dish”. In the future, we will incorporate similarity scores based on WordNet, e.g., Leacock and Chodorow’s (1998) scores for approximate lexical matches; such matches occurred in 4% of our corpus (Section 4).

To reduce the complexity of reference resolution across a sequence of sentences, and the amount of data required to reliably estimate probabilities (Section 3), we separate our problem into two parts: (1) identifying the sentence being referred to, and (2) determining the referent within that sentence.

**Identifying a sentence.** Most referents in our corpus appear in the *current, previous* or *first* sentence in a sequence, with a few referents appearing in *other* sentences (Section 4). Hence, we have chosen the sentence classes {*current, previous, first, other*}. The probability of referring to a sentence of a particular class from a sentence in position $i$ is estimated from our corpus, where $i = 1, \ldots, 5, > 5$ (there are only 13 sequences with more than 5 sentences). We estimate this distribution for each leave-one-out cross-validation fold in our evaluation (Section 4).

**Determining a referent.** We use heuristics based on those described in (Lappin and Leass, 1994) to classify pronouns (an example of a non-pronoun usage is “*It* is ModalAdjective that S”), and heuristics based on the results obtained in (Ng et al., 2005) to classify one-anaphora (an example of a high-performing feature pattern is “*one* as head-noun with NN or CD as Part-of-speech and no attached *of* PP”). If a term is classified as a pronoun or one-anaphor, then a list of potential referents is constructed using the head nouns in the target sentence. We use the values in (Lappin and Leass, 1994) to assign a score to each anaphor-referent pair according to the grammatical role of the referent in the target UCG (obtained from the highest probability parse tree that is a parent of this UCG). These scores are then converted to probabilities using a linear mapping function.

## 3 Estimating the Probability of a Merged Interpretation

We now present our formulation for estimating the probability of a sequence of UCGs, which supports the selection of the most probable sequence.

**One sentence.** The probability of a UCG generated from a sentence $T$ is estimated as described in (Zukerman et al., 2008), resulting in

$$\Pr(U|T) \propto \textstyle\sum_P \Pr(P|T)\cdot\Pr(U|P) \qquad (1)$$

where $T$, $P$ and $U$ denote text, parse tree and UCG respectively. The summation is taken over all possible parse trees from the text to the UCG, because a UCG can have more than one ancestor. As mentioned above, the parser returns an estimate of $\Pr(P|T)$; and $\Pr(U|P) = 1$, since the process of generating a UCG from a parse tree is deterministic.

**A sentence sequence.** The probability of an interpretation of a sequence of sentences $T_1, \ldots, T_n$ is

$$\Pr(U_1, \ldots, U_m|T_1, \ldots, T_n) = \\ \Pr(U_1, \ldots, U_n, M_1, \ldots, M_n, R_1, \ldots, R_n|T_1, \ldots, T_n)$$

where $m$ is the number of UCGs in a merged sequence.

By making judicious conditional independence assumptions, and incorporating parse trees into the formulation, we obtain

$$\Pr(U_1, \ldots, U_m|T_1, \ldots, T_n) = \\ \prod_{i=1}^{n} \Pr(U_i|T_i)\cdot\Pr(M_i|P_i, T_i)\cdot\Pr(R_i|P_1, \ldots, P_i)$$

This formulation is independent of the number of UCGs in a merged sequence generated by Algorithm 1, thereby supporting the comparison of UCG sequences of different lengths (produced when different numbers of mergers are performed).

$\Pr(U_i|T_i)$ is calculated using Equation 1, and $\Pr(M_i|P_i, T_i)$ is obtained as described in Section 2.2 (recall that the input features to the classifier depend on the parse tree and the sentence). In principle, $\Pr(M_i|P_i, T_i)$ and $\Pr(R_i|P_1, \ldots, P_i)$ could be obtained by summing over all parse trees, as done in Equation 1. However, at present we use the highest-probability parse tree to simplify our calculations.

To estimate $\Pr(R_i|P_1, \ldots, P_i)$ we assume conditional independence between the identifiers in a sentence, yielding

$$\Pr(R_i|P_1, \ldots, P_i) = \prod_{j=1}^{k_i} \Pr(R_{ij}|P_1, \ldots, P_i)$$

where $k_i$ is the number of identifiers in sentence $i$, and $R_{ij}$ is the referent for identifier $j$ in sentence $i$. As mentioned in Section 2.3, this factor is separated into determining a sentence, and determining a referent in that sentence. We also include in our formulation the Type of the identifier (pronoun, one-anaphor or NP) and sentence position $i$, yielding

$$\Pr(R_{ij}|P_1,\ldots,P_i) =$$
$$\Pr(R_{ij} \text{ ref } NP_a \text{ in sent } b, \text{Type}(R_{ij})|i, P_1,\ldots,P_i)$$

After additional conditionalization we obtain

$$\Pr(R_{ij}|P_1,\ldots,P_i) =$$
$$\Pr(R_{ij} \text{ ref } NP_a|R_{ij} \text{ ref sent } b,\text{Type}(R_{ij}),P_i,P_b)\times$$
$$\Pr(R_{ij} \text{ ref sent } b|\text{Type}(R_{ij}),i)\times\Pr(\text{Type}(R_{ij})|P_i)$$

As seen in Section 2.3, $\Pr(\text{Type}(R_{ij})|P_i)$ and $\Pr(R_{ij} \text{ ref } NP_a|R_{ij} \text{ ref sent } b,\text{Type}(R_{ij}),P_i,P_b)$ are estimated in a rule-based manner, and $\Pr(R_{ij} \text{ ref sent } b|\text{Type}(R_{ij}),i)$ is estimated from the corpus (recall that we distinguish between sentence classes, rather than specific sentences).

## 4 Evaluation

We first describe our experimental set-up, followed by our results.

### 4.1 Experimental set-up

We conducted a web-based survey to collect a corpus comprising multi-sentence requests. To this effect, we presented participants with a scenario where they are in a meeting room, and they ask a robot to fetch something from their office. The idea is that if people cannot see a scene, their instructions will be more segmented than if they can view the scene. The participants were free to decide which object to fetch, and what was in the office. There were no restrictions on vocabulary or grammatical form for the requests.

We collected 115 sets of instructions mostly from different participants (a few people did the survey more than once).[6] The sentence sequences in our corpus contain between 1 and 9 sentences, with 74% of the sequences comprising 1 to 3 sentences. Many of the sentences had grammatical requirements which exceeded the capabilities of our system. To be able to use these instruction sets in our evaluation, we made systematic manual changes to produce sentences that meet our system's grammatical restrictions (in the future, we will relax these restrictions, as required by a deployable system). Below are the main types of changes we made.

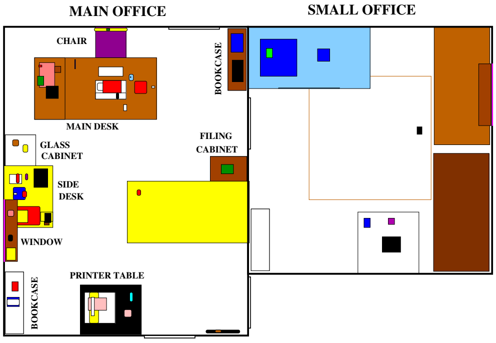


Figure 2: Our virtual environment (top view)

- Indirect Speech Acts in the form of questions were changed to imperatives. For instance, “Can you get my tea?” was changed to “Get my tea”.
- Conjoined verb phrases or sentences were separated into individual sentences.
- Composite verbs were simplified, e.g., “I think I left it on” was changed to “it is on”, and out-of-vocabulary composite nouns were replaced by simple nouns or adjectives, e.g., “the diary is A4 size” to “the diary is big”.
- Conditional sentences were removed.

Table 1 shows two original texts compared with the corresponding modified texts (the changed portions in the originals have been italicized).

Our evaluation consists of two experiments: (1) ICGs for sentence pairs, and (2) UCGs for sentence sequences.

**Experiment 1.** We extracted 106 sentence pairs from our corpus — each pair containing one declarative and one imperative sentence. To evaluate the ICGs, we constructed a virtual environment comprising a main office and a small office (Figure 2). Furniture and objects were placed in a manner compatible with what was mentioned in the requests in our corpus; distractors were also placed in the virtual space. In total, our environment contains 183 instantiated concepts (109 office and household objects, 43 actions and 31 relations). The $(x, y, z)$ coordinates, colour and dimensions of these objects were stored in a knowledge base. Since we have two sentences and their mode is known, no corpus-based information is used for this experiment, and hence no training is required.

[6] We acknowledge the modest size of our corpus compared to that of some publicly available corpora, e.g., ATIS. However, we must generate our own corpus since our task differs in nature from the tasks where these large corpora are used.

| | |
|---|---|
| **Original** | Get my book *"The Wizard of Oz"* from my office. It's green *and yellow. It has a picture of a dog and a girl on it.* It's in my *desk* drawer on the right *side* of my desk*, the second drawer down. If it's not there, it's somewhere on my shelves that are on the left side of my office as you face the window.* |
| **Modified** | Get my book from my office. It's green. It's in my drawer on the right of my desk. |
| **Original** | *DORIS, I left* my mug in my office *and I want a coffee. Can you* go into my office *and* get my mug. It is on top of the cabinet *that is* on the left *side* of my desk. |
| **Modified** | My mug is in my office. Go into my office. Get my mug. It is on top of the cabinet on the left of my desk. |

Table 1: Original and modified text

**Experiment 2.** Since UCGs contain only syntactic information, no additional setup was required. However, for this experiment we need to train our mode classifier (Section 2.2), and estimate the probability distribution of referring to a particular sentence in a sequence (Section 2.3). Owing to the small size of our corpus, we use leave-one-out cross validation.

For both experiments, *Scusi?* was set to generate up to 300 sub-interpretations (including parse trees, UCGs and ICGs) for each sentence in the test-set; on average, it took less than 1 second to go from a text to a UCG. An interpretation was deemed successful if it correctly represented the speaker's intention, which was represented by an imperative Gold ICG for the first experiment, and a sequence of imperative Gold UCGs for the second experiment. These Gold interpretations were manually constructed by the authors through consensus-based annotation (Ang et al., 2002). As mentioned in Section 2, we evaluated only imperative ICGs and UCGs, as they contain the actions the robot is expected to perform.

### 4.2 Results

Table 2 summarizes our results. Column 1 shows the type of outcome being evaluated (ICGs in Experiment 1, and UCG sequences and individual UCGs in Experiment 2). The next two columns display how many sentences had Gold interpretations whose probability was among the top-1 and top-3 probabilities. The average *rank* of the Gold interpretation appears in Column 4 ("not found" Gold interpretations are excluded from this rank). The rank of an interpretation is its position in a list sorted in descending order of probability (starting from position 0), such that all equiprobable interpretations have the same position. Columns 5 and 6 respectively show the median and 75%-ile rank of the Gold interpretation. The number of Gold interpretations that were not found appears in Column 7, and the total number of requests/UCGs is shown in the last column.

**Experiment 1.** As seen in the first row of Table 2, the Gold ICG was top ranked in 75.5% of the cases, and top-3 ranked in 85.8%. The average rank of 2.17 is mainly due to 7 outliers, which together with the "not-found" Gold ICG, are due to PP-attachment issues, e.g., for the sentence pair "Fetch my phone from my desk. It is near the keyboard.", the top parses and resultant UCGs have "near the keyboard" attached to "the desk" (instead of "the phone"). Nonetheless, the top-ranked interpretation correctly identified the intended object and action in 5 of these 7 cases. Median and 75%-ile results confirm that most of the Gold ICGs are top ranked.

**Experiment 2.** As seen in the second row of Table 2, the Gold UCG sequence was top ranked for 51.3% of the requests, and top-3 ranked for 53.0% of the requests. The third row shows that 62.4% of the individual Gold UCGs were top-ranked, and 65.4% were top-3 ranked. This indicates that when *Scusi?* cannot fully interpret a request, it can often generate a partially correct interpretation. As for Experiment 1, the average rank of 3.14 for the Gold UCG sequences is due to outliers, several of which were ranked above 30. The median and 75%-ile results show that when *Scusi?* generates the correct interpretation, it tends to be highly ranked.

Unlike Experiment 1, in Experiment 2 there is little difference between the top-1 and top-3 results. A possible explanation is that in Experiment 1, the top-ranked UCG may yield several probable ICGs, such that the Gold ICG is not top ranked — a phenomenon that is not observable at the UCG stage.

Even though Experiment 2 reaches only the

Table 2: *Scusi?*'s interpretation performance

| | **# Gold interps. top 1** | **with prob. in top 3** | **Average rank** | **Median rank** | **75%-ile rank** | **Not found** | **Total #** |
|---|---|---|---|---|---|---|---|
| **ICGs** | 80 (75.5%) | 91 (85.8%) | 2.17 | 0 | 0 | 1 (0.9%) | 106 reqs. |
| **UCG seqs.** | 59 (51.3%) | 61 (53.0%) | 3.14 | 0 | 1 | 36 (31.3%) | 115 reqs. |
| **UCGs** | 146 (62.4%) | 153 (65.4%) | NA | NA | NA | 55 (23.5%) | 234 UCGs |

UCG stage, *Scusi?*'s performance for this experiment is worse than for Experiment 1, as there are more grounds for uncertainty. Table 2 shows that 31.3% of Gold UCG sequences and 23.5% of Gold UCGs were not found. Most of these cases (as well as the poorly ranked UCG sequences and UCGs) were due to (1) imperatives with object specifications (19 sequences), (2) wrong anaphora resolution (6 sequences), and (3) wrong PP-attachment (6 sequences). In the near future, we will refine the merging process to address the first problem. The second problem occurs mainly when there are multiple anaphoric references in a sequence. We propose to include this factor in our estimation of the probability of referring to a sentence. We intend to alleviate the PP-attachment problem, which also occurred in Experiment 1, by interleaving semantic and pragmatic interpretation of prepositional phrases as done in (Brick and Scheutz, 2007). The expectation is that this will improve the rank of candidates which are pragmatically more plausible.

## 5 Related Research

This research extends our mechanism for interpreting stand-alone utterances (Zukerman et al., 2008) to the interpretation of sentence sequences. Our approach may be viewed as an *information state* approach (Larsson and Traum, 2000; Becker et al., 2006), in the sense that sentences may update different informational aspects of other sentences, without requiring a particular "legal" set of dialogue acts. However, unlike these information state approaches, ours is probabilistic.

Several researchers have investigated probabilistic approaches to the interpretation of spoken utterances in dialogue systems, e.g., (Pfleger et al., 2003; Higashinaka et al., 2003; He and Young, 2003; Gorniak and Roy, 2005; Hüwel and Wrede, 2006). Pfleger *et al.* (2003) and Hüwel and Wrede (2006) employ modality fusion to combine hypotheses from different analyzers (linguistic, visual and gesture), and apply a scoring mechanism to rank the resultant hypotheses. They disambiguate referring expressions by choosing the first object that satisfies a 'differentiation criterion', hence their system does not handle situations where more than one object satisfies this criterion. He and Young (2003) and Gorniak and Roy (2005) use Hidden Markov Models for the ASR stage. However, these systems do not handle utterance sequences. Like *Scusi?*, the system developed by Higashinaka *et al.* (2003) maintains multiple interpretations, but with respect to dialogue acts, rather than the propositional content of sentences. All the above systems employ semantic grammars, while *Scusi?* uses generic, statistical tools, and incorporates semantic- and domain-related information only in the final stage of the interpretation process. This approach is supported by the findings reported in (Knight et al., 2001) for relatively unconstrained utterances by users unfamiliar with the system, such as those expected by *DORIS*.

Our mechanism is also well suited for processing replies to clarification questions (Horvitz and Paek, 2000; Bohus and Rudnicky, 2005), since a reply can be considered an additional sentence to be incorporated into top-ranked UCG sequences. Further, our probabilistic output can be used by a utility-based dialogue manager (Horvitz and Paek, 2000).

## 6 Conclusion

We have extended *Scusi?*, our spoken language interpretation system, to interpret sentence sequences. Specifically, we have offered a procedure that combines the interpretations of the sentences in a sequence, and presented a formalism for estimating the probability of the merged interpretation. This formalism supports the comparison of interpretations comprising different numbers of UCGs obtained from different mergers.

Our empirical evaluation shows that *Scusi?* performs well for textual input corresponding to (modified) sentence pairs. However, we still need to address some issues pertaining to the integration of UCGs for sentence sequences of arbitrary length. Thereafter, we propose to investigate the influence of speech recognition performance on *Scusi?*'s performance. In the future, we intend to expand *Scusi?*'s grammatical capabilities.

## Acknowledgments

This research was supported in part by grant DP0878195 from the Australian Research Council.

## References

J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke. 2002. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *ICSLP'2002 – Proceedings of the 7th International Conference on Spoken Language Processing*, pages 2037–2040, Denver, Colorado.

T. Becker, P. Poller, J. Schehl, N. Blaylock, C. Gerstenberger, and I. Kruijff-Korbayová. 2006. The SAMMIE system: Multimodal in-car dialogue. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 57–60, Sydney, Australia.

D. Bohus and A. Rudnicky. 2005. Constructing accurate beliefs in spoken dialog systems. In *ASRU'05 – Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 272–277, San Juan, Puerto Rico.

T. Brick and M. Scheutz. 2007. Incremental natural language processing for HRI. In *HRI 2007 – Proceedings of the 2nd ACM/IEEE International Conference on Human-Robot Interaction*, pages 263–270, Washington, D.C.

P. Gorniak and D. Roy. 2005. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *ICMI'05 – Proceedings of the 7th International Conference on Multimodal Interfaces*, pages 138–143, Trento, Italy.

Y. He and S. Young. 2003. A data-driven spoken language understanding system. In *ASRU'03 – Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 583–588, St. Thomas, US Virgin Islands.

R. Higashinaka, M. Nakano, and K. Aikawa. 2003. Corpus-Based discourse understanding in spoken dialogue systems. In *ACL-2003 – Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 240–247, Sapporo, Japan.

E. Horvitz and T. Paek. 2000. DeepListener: Harnessing expected utility to guide clarification dialog in spoken language systems. In *ICSLP'2000 – Proceedings of the 6th International Conference on Spoken Language Processing*, pages 229–229, Beijing, China.

S. Hüwel and B. Wrede. 2006. Spontaneous speech understanding for robust multi-modal human-robot communication. In *Proceedings of the COLING/ACL Main Conference Poster Sessions*, pages 391–398, Sydney, Australia.

S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. 2001. Comparing grammar-based and robust approaches to speech understanding: A case study. In *Proceedings of Eurospeech 2001*, pages 1779–1782, Aalborg, Denmark.

S. Lappin and H.J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20:535–561.

S. Larsson and D. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340.

C. Leacock and M. Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–285. MIT Press.

H.T. Ng, Y. Zhou, R. Dale, and M. Gardiner. 2005. A machine learning approach to identification and resolution of one-anaphora. In *IJCAI-05 – Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1105–1110, Edinburgh, Scotland.

N. Pfleger, R. Engel, and J. Alexandersson. 2003. Robust multimodal discourse processing. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue*, pages 107–114, Saarbrücken, Germany.

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

I. Zukerman, E. Makalic, M. Niemann, and S. George. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, pages 581–592, Hanoi, Vietnam.

G. Zweig, D. Bohus, X. Li, and P. Nguyen. 2008. Structured models for joint decoding of repeated utterances. In *Proceedings of Interspeech 2008*, pages 1157–1160, Brisbane, Australia.