Possibilities for Enhancing Speech Recognition by Consulting Information States

J. F. Quesada J. G. Amores P. Manchón G. Pérez
S. Knight D. Milward J. Thomas

Distribution: PUBLIC

Specification, Interaction and Reconfiguration in Dialogue Understanding Systems
IST-1999-10516

Deliverable D2.3
October, 2002

IST-1999-10516 SIRIDUS

Specification, Interaction and Reconfiguration in Dialogue Understanding Systems

Göteborg University
Department of Linguistics

Linguamatics Ltd

Telefónica Investigación y Desarrollo SA Unipersonal
Speech Technology Division

Universität des Saarlandes
Department of Computational Linguistics

Universidad de Sevilla
Departamento de Lengua Inglesa

For copies of reports, updates on project activities and other SIRIDUS-related information, please look on our website at www.ling.gu.se/projekt/siridus.

For technical matters, please contact the Technical Coordinator, Robin Cooper and for administrative matters, please contact the Administrative Coordinator, Mareike Schmitt.

Prof. Robin Cooper
Department of Linguistics
Gothenburg University
Box 200
SE-405 30 Gothenburg
Sweden

Phone: +46 31 773 2536
Fax: +46 31 773 4853
cooper@ling.gu.se

Mareike Schmitt
European Project Office
Saarland University
c/o EURICE GmbH
Science Park Saar
Stuhlsatzenhausweg 69
D-66123 Saarbrucken
Germany
Phone: +49 (0) 681 959 233 66
Fax: +49 (0) 681 959 233 70
ms@eurice.de

Contents

Chapter 1

Introduction

It has long been recognised that recognition results can be improved by taking account of the context in which an utterance is spoken. In system initiated dialogues this is usually achieved by having separate language models associated with different prompts. However, this solution does not carry over easily to flexible dialogue systems. Firstly, language models tend to be much larger to deal with the greater variety of user utterances. Swapping language models therefore becomes much slower. Secondly, language models are often statistically based (e.g. using n-grams), so separate language models per prompt are expensive to collect. Thirdly, the number or prompts may be extremely large. In fact, if generation is particularly advanced, the number of possible prompts could actually be infinite.

In this deliverable we present two approaches which keep the language models constant, whatever the context, but provide an opportunity for the information state to affect a choice between alternative recognition hypotheses. In Chapter 2 we show how the information state can improve the choice of hypothesis from a list of the top 'n' recogniser hypotheses. In Chapter 3 we outline some of the building blocks required for choosing hypotheses directly from the recogniser word lattice. Results in this deliverable are promising, but very preliminary.

Chapter 2

Using context to choose between hypotheses

2.1 Introduction

One way of utilising information state or any kind of dialogue knowledge to improve performance is to obtain an n-best list from a speech recogniser, and choose the final interpretation from that list rather than taking the highest-scoring (or 1st-best) hypothesis from the recogniser. This choice can be made using weightings based on semantic plausibility of the utterance and current dialogue state, e.g. [Van Noord *et al*, 1999, Koeling,2002].

In this chapter we will discuss ways in which Information State (IS) information can be used to choose between hypotheses in a dialogue system for home control built at Linguamatics. The user communicates with the house through speech (via Nuance v8.0, using speaker independent, statistical language models) or by clicking on icons which represent the devices in the house in a Java interface. The system updates the house representation and interacts with the user via voice synthesis.

In this domain, the IS could contain knowledge of which devices are present in which rooms in the house, and whether they are switched on and off (or are in some other state, e.g. the TV can be showing BBC1, BBC2 and so on; the front door can be locked or unlocked; the curtains can be open or shut.) The statistical probability assigned by the recogniser is available, although this unfortunately differs little between the top and lower-ranked interpretations, for many utterances. Possibly more informative might be the word probabilities assigned to the semantically-rich words, also available from Nuance. These probabilities can be combined with IS-based preferences, such as a low probability for a

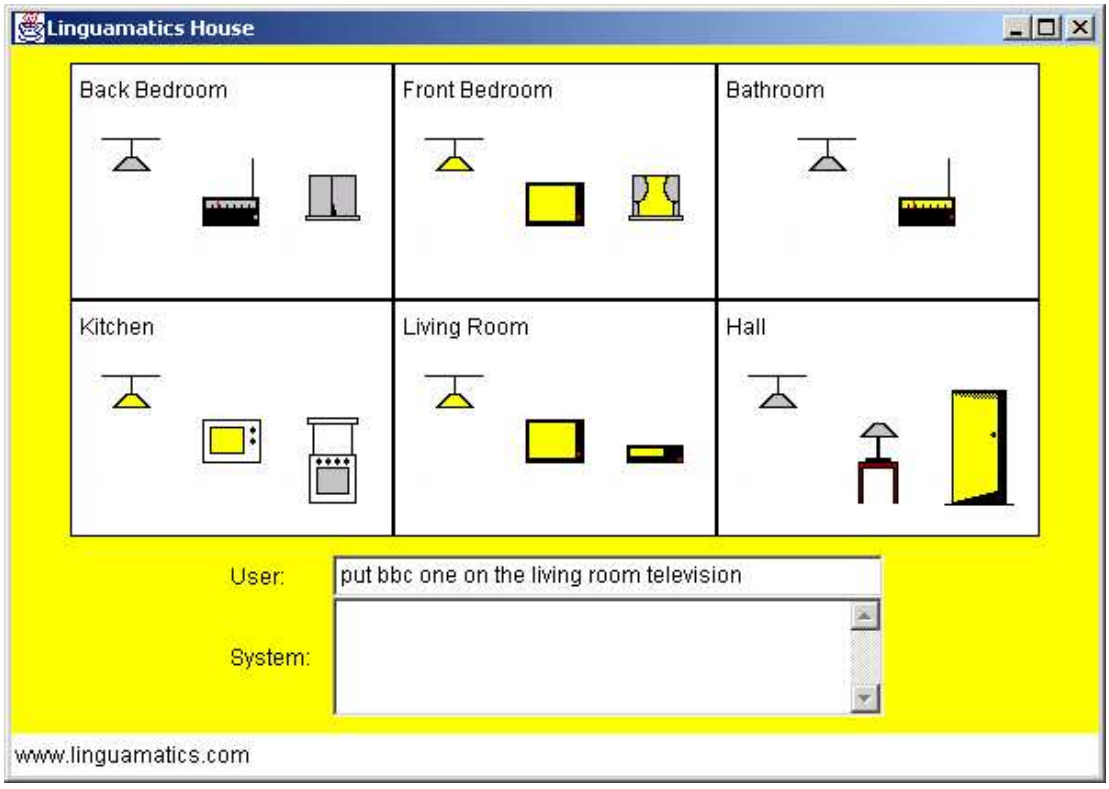

Figure 2.1: The Linguamatics House Simulator

device-room pair that does not exist[1]. The recogniser often confuses "on" and "off", and a useful semantic preference might boost the interpretation corresponding to the current state of the device, i.e. preferring "switch on" to "switch off" when the device is currently off.

2.2 The House Simulator

Figure 2.1 shows the House Simulator. The underlying architecture of the system is given by the diagram in Figure 2.2. Speech recognition and synthesis are controlled by a process, Speech Controller, which communicates with the Dialogue Manager (DM) and the Java House Simulator through a simple message-passing router.

Interaction between the user and the system is multi-modal. The user may speak or click

[1] Unless in the particular form "is there a heater in the lounge."

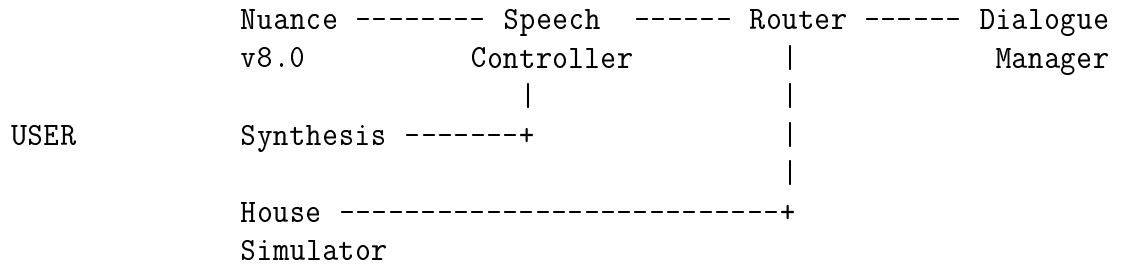

Figure 2.2: Linguamatics House System Architecture

on devices in the house to simulate turning them on and off. All interaction triggers a message to the dialogue manager which in turn sends commands to the house (specifically, to devices in the house) and the synthesiser to reflect any updates to the state of the house, e.g. that the living room light is now on.

To analyse the performance of the Dialogue Manager, we replace the Speech Controller with a testing module which feeds input to the DM as if it were coming from Nuance and can access all messages passed by the router (Figure 2.3.) By logging the DM messages we are interested in, we are able to compare actual to expected behaviour.

The House Simulator is still required in the testing scenario since, although there is no user to interact with, the Dialogue Manager has the ability to poll devices to determine their current status.

2.3 Choosing an Evaluation Methodology

We can envisage several ways to evaluate the Dialogue Manager from word and utterance error rates through to semantic analysis error rates or task completion scores.

Nuance provides a mechanism (a program called `batchrec`) for generating word and sentence error rates of a particular language model against a corpus of .wav files and their transcriptions. Unfortunately, this testing does not permit the interleaving of calls to the Dialogue Manager and so it is not possible to generate comparative statistics for systems with and without a DM in this way.

However, it is possible to use some of the information input to and generated by `batchrec`

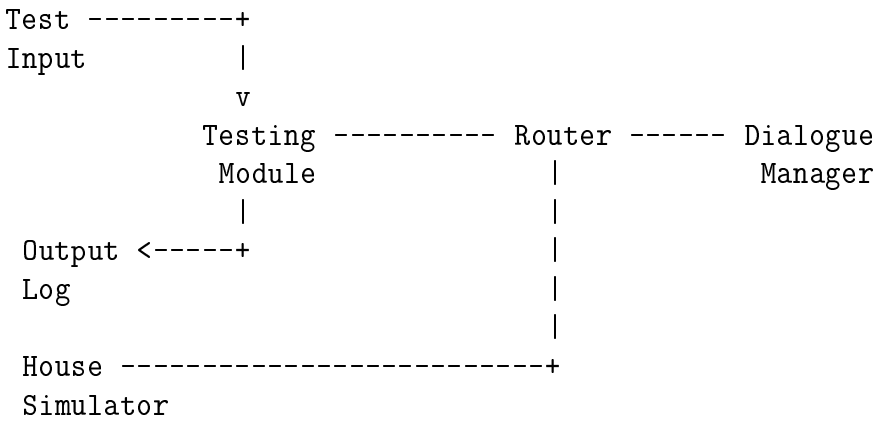

Figure 2.3: Linguamatics House System Architecture for Testing

in a simple (and relatively simple-minded) way. This method feeds the n-best lists recorded by `batchrec` as the output of the language models on a particular corpus into the Dialogue Manager and compares the DM's preferred utterances against the file of transcriptions.

This technique gives only yes/no statistics on whether the DM's preferred utterance exactly matches the transcription and so in a sense is quite harsh since it takes no account of the similarity of the chosen and actual utterance (e.g. there is no word error rate.)

Work on dialogue evaluation in projects such as PARADISE and COMMUNICATOR (e.g. [Walker *et al*, 2000, Walker *et al*, 1997]) has put forward various scores for evaluating the performance of dialogue managers and dialogue systems which take account of the shortcomings of simply matching against "reference answers."

Danieli and Gerbino [Danieli & Gerbino, 1995] suggest an Implicit Recovery metric which would take account of the capability of a DM to "regain utterances which are partially failed at recognition or understanding levels." This requires a semantic representation which enables the evaluation of the number of "concepts" which are correct for an utterance. Something along these lines would certainly remove the "noise" that a purely word-based evaluation gives, but is difficult to undertake at this stage since the semantics of the current DM are relatively simple and liable to change and, indeed, may be distributed throughout the DM rather than being a single set of (say) first-order expressions. This method of evaluation would also still suffer from not crediting the DM with correct (robust) behaviour in situations where the chosen semantics are not identical to the transcript but the DM carries out the (or an) appropriate and/or sensible action.

Evaluation in the PARADISE and COMMUNICATOR projects start from the position that "user satisfaction is the overall objective to be maximized and that task success and various interaction costs can be used as predictors of satisfaction." [Walker *et al*, 2000]. In extended dialogues for tasks such as airline booking, this evaluation typically attempts to quantify the quality of the dialogue in some way. These measures are intended to credit the DM with reasonable behaviour, e.g. a clarification in cases where it is not confident, but penalise it for inappropriate behaviour, e.g. an excessive number of dialogue turns to complete a simple task. The over-riding concern, however, is always that the user's required task be completed.

For initial evaluation, the most promising option for reasonable, repeatable, automated evaluation is matching on the DM's choice of utterance from the n-best list. This will allow us to rapidy evaluate DMs on different corpora, but we will need to be aware of its shortcomings. Our baseline score will be the percentage of 1st-best utterances which are also the actual utterance, i.e. we will be comparing the DM plus IS against a simple strategy which merely chooses the first utterance on the n-best list provided by the speech recogniser.

2.4 Evaluation

Although we hope to improve the performance of the DM in general dialogue, we expect that the Information State will be of particular use in specific contexts, such as clarification dialogues. Evaluating in general dialogue does not require any special machinery beyond what has already been discussed. Testing in specific contexts requires some way of setting up the context in which a particular utterance can be evaluated.

The Testing Module in Figure 2.3 can read in not only a set of utterances (or n-best lists) to feed to the DM as if they came from Nuance, but also sets of commands which do not cause anything to be recorded in the output log, but which set up the state of the house ready for a particular utterance. For instance, Figure 2.4 shows a fragment of an input file for testing clarifications like U2 in dialogues such as:

U1: Turn the bedroom light on.
S1: Which bedroom?
U2: The back bedroom.

In these examples, to make the testing code generic (i.e. not tied to a particular house), we do not control the devices in the house directly, but instead use the DM on commands

```
c turn all the radios on
c turn the microwave off
c turn the radio off
Batchrec_testing/corpora/SpecD/02_09_10_12_51_07.wav is the radio in
the bathroom, the radio in the bathroom,is the radio and the
bathroom,is there a radio in the bathroom,the radio and the
bathroom,dim radio and the bathroom,is the radios in the bathroom,the
radios in the bathroom,are the radio in the bathroom,radio in the
bathroom

c turn all the televisions off
c turn the microwave off
c turn the t v to channel four
Batchrec_testing/corpora/GeneralP/02_09_09_06_58_12.wav the back
bedroom light,the back bedroom lights,is the back bedroom light,is the
back bedroom lights,the back bedroom and the light,and the back
bedroom light,and the back bedroom lights,are the back bedroom
light,are the back bedroom lights,is the back bedroom and the light
```

Figure 2.4: Input file for testing clarifications

which we know it treats correctly to set up the house context. For example, the first set of instructions (labelled with an initial "c" in Figure 2.4) turn all of the radios in the house on, set a "null" context by mentioning an unrelated device and then (simulating U1 in a dialogue like the one above) ask to turn an unspecified radio off. The next utterance is the user response (as an n-best list) which is to be evaluated. We record the utterance that the DM chooses and can then compare this to the actual transcription.

2.4.1 Dialogue Manager

The Dialogue Manager uses semantic-based composition strategy as described in D4.4 [Milward, 2002]. It was designed for putting together fragmentary input and currently does not perform any syntactic analysis, instead concentrating on relational semantics.

Corpus	#Utterances
GeneralX	398
GeneralD	72
GeneralL	33
GeneralP	77
SpecD	65
ClarifX	35

Key: General - general, Spec - specific, Clarif - clarifications
D - Desktop, L - Laptop, P - Phone, X - any source

Table 2.1: Test Corpora

2.4.2 Test Data

We used several sets of test data, collected on different computers in different circumstances with different microphones and all including data from assorted speakers. These are summarised in Table 2.1. The "D" suffix corpora were recorded with a high-quality Sony microphone on a desktop computer; the "L" suffix on a laptop with a cheap, low-quality, but small and portable, unbranded mic; the "P" suffix corpus was gathered over the phone and the "X" suffix denotes a mixture of sources.

The "General" corpora were gathered during development and testing of this and earlier systems — as recommended by Knight et al [Knight *et al*, 2001] — and so includes some utterances from clients, family members, friends and so on. The "Clarif" corpus was partially extracted from the General corpora and contains examples of clarification dialogues. To boost the size of this corpus, the "Spec" corpus was created specifically to generate clarifications, but also includes the utterances which set up the context for a clarification.[2]

2.4.3 Examination of the Data

The obvious question to ask is where the speech recognition typically fails. Across the General and Spec corpora (the Clarif corpus is an extraction from these) there are 79 cases where the actual utterance is in the n-best list but is not the 1st best. 79 utterances is around 12% of the corpus. There are additionally 67 (11%) actual utterances which do not appear in the nbest list at all which means that 77% of all utterances are correctly recognised in the 1-best position.

[2]We are interested in whether the distribution of cases where IS could be exploited is distributed evenly across these corpora or whether different microphones (including the telephone) cause a different distribution of speech recognition errors.

Error	#Occurrences
plural	18
off/on	13
synonyms	11
initial is/and	8
and/in	7
it/them	1
a/the	1
plausible higher	10
no plausible higher	5
other	6

Table 2.2: Preliminary examination of the data

With recognition of this quality we must be careful to apply the IS knowledge appropriately. Table 2.2 shows how the actual utterance differs from the 1st best[3] in the 79 cases and we discuss below how to identify them and how IS can help to choose the actual utterance from the n-best list.

Singular/plural pairs

In 18 cases[4] the actual and 1st-best utterances differ only in that a device has been recognised as a singular or plural occurrence, e.g. "the radio" versus "the radios."

An Information State could disambiguate where possible by checking to see whether there actually are multiple occurrences. Additionally, if the location of the user is known it can also be used to add weight to a hypothesis. For instance if the device is "television" and the user is in a room with a single television, this adds an extra weight to the singular interpretation providing there is no locative modifier in the utterance.

[3]One of the results is counted twice because it was both plural and and/in.

[4]19 cases if we include the "it"/"them" example in which two utterances differ only in the number of the pronoun.

Off/on pairs

The off/on cases are pairs of sentences which differ only in the occurrence of "on" or "off": "turn the kitchen lights off" versus "turn the kitchen lights on." Again the state of the house can be used to disambiguate fairly straightforwardly. Combining this with recognition scores from Nuance would also provide a way to provide an appropriate range of responses – from simply performing the required action when the requested state is different from the actual state, to reporting an error ("the light is already on") when the states are the same and recognition score is high and entering a clarification dialogue when the states are the same and recognition is low ("did you want to turn the light off?")

Examining the data further reveals that utterances ending in "off" are typically recognised with greater confidence than those ending in "on".[5] For instance, in 84% of cases where the 1st-best is also the actual utterance, the corresponding "on" utterance does not appear anywhere in the n-best. The opposite case is starkly different: in only 20% of cases does the corresponding "off" utterance not appear.

This is an interesting result as it means that it would be possible to create different strategies for using the IS in cases where the 1st-best utterance is "on" and "off," perhaps giving more weight to the recognition scores if they indicated an "off"-utterance. However, this is a very specific measure, not generalisable and possibly would even become inapplicable with a different set of speakers, microphone and so on.

Synonymous utterances

The 11 "synonymous utterances" include pairs like "switch the lights off" and "turn the lights off" and "what is on" versus "what's on." Since we expect synonymous utterances to be interpreted and acted upon the same way, effort made in trying to choose the actual utterance for the purposes of this trial will probably be wasted in the long term.

Spurious initials

There are 8 cases of a spurious initial "is" or "and" e.g. "is the back bedroom light" as against "the back bedroom light." These tend to occur in the clarification corpus where they are the user response to a system question like "which light?"

The dialogue state will form part of any Information State, so we expect to be able to boost

[5]Note that our SLM is class-based, with a class for "on" and "off" which weights them both equally.

performance on the response to clarification questions. In particular, the spurious initial short words that the recogniser picks up in these contexts can be ruled out purely on the basis of dialogue context, unless they form part of a coherent and meaningful utterance: "is the back bedroom light on?"

and/in pairs

The other large class is "and"/"in" examples. This includes pairs such as "the light in the kitchen" and "the light *and* the kitchen." A syntactic analysis would not be enough to prefer one or the other since both utterances are syntactically correct, so we would need to rely on semantics. (As it happens, our DM is robust enough to behave correctly, even when the "and" case is chosen from the n-best.)

No plausible higher n-best

There are 5 cases where no syntactically plausible example is higher in the n-best list than the actual utterance. The Dialogue Manager does not currently perform a syntactic analysis of input utterances. so syntax alone should be enough to obtain these.

Plausible higher n-best

There are 10 examples where syntactically plausible utterances are higher in the n-best list than the actual utterance. The IS could use knowledge about the state of the house to rule out the syntactically plausible, but physically incorrect, possible utterances. For instance, in the following n-nbest list the correct utterance is 2nd but the 1st-best is very plausible unless the oven is already off, in which case some kind of clarification dialogue could be entered into.

```
1. turn the oven in the kitchen off
2. turn everything in the kitchen off
3. turn oven in the kitchen off
4. is there a oven in the kitchen off
5. turn the fan in the kitchen off
```

Similarly, in the following list the 3rd utterance is correct but the 1st is plausible as long as there is a hi-fi in the kitchen.

```
1. turn on the hi fi in the kitchen
2. turn on the oven on in the kitchen
3. turn on everything in the kitchen
4. turn on the cooker in the kitchen
5. turn on the fan on in the kitchen
```

This final example is harder, since there are three plausible utterances (2, 3, 4) higher up the n-best list than the actual utterance (6.)

```
1. which light are off
2. which lights are off
3. put the light off
4. put the lights off
5. which device are off
6. is the light off
7. is the lights off
```

Other

There are a handful of cases where the transcription is odd, usually because the recogniser cut off either the beginning or the end of a user utterance. The correct approach for the dialogue manager when encountering utterances of this type is a clarification dialogue.

2.5 Improving Speech Recognition with the IS

Given the evaluation strategy outlined in Section 2.3 the obvious baseline against which to measure performance is the percentage of utterances where the 1st-best utterance from the recogniser matched the transcription. We can also calculate a potential gain by looking at the percentage of utterances where the transcription appeared somewhere in the n-best list, i.e. where the actual utterance is available for selection from the n-best list. Similar work, inspired by Ginzburg's approach to dialogue, is outlined in [Van Noord *et al*, 1999, Koeling,2002].

2.5.1 Clarification Contexts

Having already identified clarification dialogues as a potential target for IS exploitation, we initially focussed on the ClarifX corpus. Section 2.4.3 records that "spurious initials" are a problem in this data set—it is the case that the statistical language models prefer utterances which begin with "is" to those that begin with "the" even if the acoustic scores do not suggest the latter. Users tend to respond with "the living room TV" when faced with a question such as "which TV?"

It is possible to adjust various parameters in Nuance to give more weight to acoustic models over the trigram and bigram statistics. The `batchrec` program already mentioned provides an objective way to evaluate the effectiveness of different settings and we are already using a set of parameters which gives the best overall performance on the GeneralX corpus. Reducing reliance on the n-gram statistics can be useful when an SLM is poorly trained, but in a well-trained model, it tends to increase the word error rate.

It is also possible to provide more examples of NP utterances ("the living room," "the bedroom television," "the front door" etc) so that the statistical preference for "is the living room" with a spurious "is" is lessened. This approach is something of an inexact science, but is certainly a plausible route although it is possible that it will reduce performance in some other area of the system.

Both of these potential corrections rely on specialising the language models to the domain and hence they, or procedures akin to them, will be required in every domain that the dialogue manager is applied to. It is certainly the case that in any domain we will want to tune for best performance, but it is also the case that a more general solution would place extra robustness in the dialogue manager, a generic linguistic solution rather than a per-case bespoke (semi) statistical one.

By adding a notion of dialogue context to the Information State, we are able to identify situations in which we expect a certain form of answer—in the first instance, in the case of clarifications about device or location, this is a simple NP but could easily become a more complex description with type information e.g. `NP[type=room]` or `NP[type=device]` given appropriate ontological knowledge in the DM. The DM is now able to reject the 1st-best utterance when it begins with, for example, "is" (unless the whole recognised utterance forms a valid utterance in its own right, e.g. "is the living room television on") in favour of the next most highly ranked utterance which is of the expected type.

To illustrate the improvement that this particular element of the IS can give, Table 2.3 shows the baseline, potential and DM scores on our corpora, where *DM(cc)* indicates a baseline DM with clarification context in its dialogue state.[6] We can see that the ClarifX

[6]In these figures, n=10 and we do not count utterances that the speech recogniser rejected, i.e. if

Corpus	Baseline	Potential	DM(cc)
GeneralX	80.83	90.16	81.66
GeneralD	66.67	83.33	66.67
GeneralL	60.61	78.79	62.50
GeneralP	77.92	89.61	77.92
SpecD	69.23	93.85	75.00
ClarifX	60.00	85.71	74.29

Table 2.3: Evaluation of Dialogue Manager with clarification context (cc) against baseline and potential scores

corpus performance has improved dramatically, with the DM now choosing the correct utterance from the n-best list over the incorrect 1st-best utterance in all "spurious initial" cases in the corpus. Performance on other corpora has not decreased and has in some cases increased slightly. This is because this DM will choose the 1st-best utterance in all cases other than those where it is in a clarification context. In the other corpora it is possible that successive pairs of utterances form a clarification context — there is no "reset context" command between utterances — which is why there are some slight improvements.

2.5.2 Further Work

Section 2.4.3 identified areas which form the basis of ongoing research. There are interesting issues to do with the Information State architecture required to cope with the off/on and plural examples. For instance, in order to know about the state of the house, should the IS model the house or use the house itself as a model of its own state?

If we model the house state in the IS, then we must ensure that the model and the actual house state remain closely synchronised. If they do not thent here is scope for misunderstanding between the DM and user and the DM and devices in the home. Both of these will result in poorer system performance.

If we use the state of the house as its own model, there is the problem of potential delays in the house response to the DM's queries about its state. The devices might be slow to respond and the medium over which communication takes place (the house network) might be slow.

Speed is a crucial factor in the usability of dialogue systems, but duplication of information

the DM scores 80%, that is 80% of the utterances which generated an n-best list and not 80% of the utterances in the corpus. (The figures in Table 2.1 do not include utterances which are rejected by the speech recogniser.)

is generally poor practice, so some work on the trade-offs between the two approaches needs to be done.

Our Information State already contains the current context which is used for pronoun and reference resolution. Analagously to the house state information, dialogue context can also improve speech recognition by ruling out, or dispreferring, sentences which do not have a likely referent in the context.

We have also come across examples where we might plausibly claim that *less* use of the IS is required to achieve the correct interpretation—assuming that the IS only contains information about what is actually *in* the house, and user/system dialogues. For instance, we would not want the DM to disprefer a user utterance such as "is there a lava lamp in the lounge?" purely on the basis of there not being a lava lamp in the lounge. (Whereas we would like to disprefer "turn the lava lamp on" when there is no lava lamp.)

We are experimenting with these approaches and hope to have results to report before the project ends.

2.6 Conclusions

In this chapter we have discussed the ways in which an Information State can be used to improve speech recognition results by influencing the choice of utterance from an n-best list. We have spent time investigating the type of errors that simply choosing the 1st-best utterance will generate and have suggested ways in which the IS can be used to cure them. In the case of one type of error, the "spurious intials" we have extended our IS and produced a significant improvement in performance. We then introduced some of the issues which will concern our further work in this area.

We have discussed ways in which performance can be measured for this particular task and settled on a strict string-matching approach which has the advantages of being simple, quick, cheap to calculate on multiple runs of multiple data sets. On its downside, we noted that it could be more flexible, e.g. by giving credit for correct words in an incorrect sentence, and does not directly measure overall system performance.

In particular, this evaluation method does not give credit for the robustness of the dialogue manager in coping with an incorrect choice from the n-best list. In fact, parallel to this work, we did perform a higher-level analysis of the system, attempting to evaluate broadly at a semantic level how well the system performed. This work is reported in [Milward, 2002].

Chapter 3

General Technique: Integration of Speech Recognition, Semantic Interpretation and Dialogue Management

3.1 Introduction

The Spoken Dialogue System Architecture chosen for Siridus, as well as for the majority of the systems mentioned in the literature, is based on the functional distribution of the tasks at hand into a set of modules or agents.

This division facilitates the development of each of the independent modules and the re–usability of the best outputs of each of the functional modules. It thus helps all software engineering issues related to the design and implementation of these systems.

Nonetheless, from a psycholinguistic point of view, this functional distribution does not appear natural. From this point of view, it seems more reasonable to advocate for a more global approach that integrates the recognition and natural language processes, and the semantic and pragmatic interpretation of the dialogues.

It is obvious that a robust integration of all these modules or agents would imply substantial improvements in Spoken Dialogue Systems.

3.2 Background and Motivation

Currently, one of the main challenges in the design and implementation of Spoken Dialogue Systems is the integration of each of the components into the system. There is an increasing number of applications based on the integration of Speech Recognition, Speech Synthesis, Parsing, Semantic Interpretation and Dialogue Management. Within the broad range of architectures or implementations currently under development, the common factor is the use of specialized subsystems to handle each of the main phases:

- Speech Recognition
- Semantic Interpretation
- Dialogue Management
- NLG / Speech Synthesis

Breaking the process into independent subsequent phases as listed above can be easily implemented. However, each of the components will be deprived of the information made available by the rest of the components, which will have a negative impact in the final results. A tighter integration between the different modules would render better results at several levels:

- *ASR and Semantic Interpreter (SI) integration*: the SI could provide the ASR with high level linguistic information. It can guide the recognition process by discarding semantically unlikely candidates and maybe even boosting the likelihood of the more semantically plausible candidates. Given a complete system integration, the SI will also take the DM's expectations into account to help the recognition process.
- *Semantic Interpreter and Dialogue Manager*: the SI could benefit from the dialogue expectations generated by the DM. In turn, the SI will be able to provide the DM with more reliable information.
- *ASR, SI and DM*: the recognition process could also be enhanced by the dialogue expectations generated by the DM. Dialogue–context specific subgrammars could be generated on–the–fly to enhance the performance of the ASR.

There have been several attempts to improve voice recognition systems by means of parsing techniques:

- Application of parsing techniques for the selection of the most likely word sequence generated by the recogniser [Van Noord *et al*, 1999].
- Implementation of a Left–to–Right parser to extract the semantic nuclei. It was also used to improve the trigram models used to generate language models [Chelba & Jelinek, 1999].
- Khundanpur and Wu (1999) used a similar model to Chelba and Jelinek. Their model integrates n–gram structures on the nuclei within a maximum entropy frame. They concluded that:

> The use of syntactic structure in general and heads of syntactic constituents in particular has recently been shown to be beneficial for statistical language modeling. (...)It is shown that the predictive power of syntactic heads is mostly complementary to the predictive power of N–grams: they help in positions where an intervening phrase or clause separates the heads from the word being predicted, making the N–gram a poor predictor. [Khudanpur & Wu, 1999].

3.3 Speech Recognition, Semantic Interpretation and Dialogue Management Systems

In this section, we will formalize processes and components in order to analyse the main problems and solution strategies.

3.3.1 Speech Recognition

Currently, the leading approach in Speech Recognition is based on the MAP criterion (*Maximum A Posterior Probability*). This strategy optimises the posterior probability of a word sequence, given acoustic input. This strategy also entails the use of a n–gram language model obtained from a domain–dependent corpus. Based on this information, the Speech Recogniser will generate a word lattice from which the word sequence with the highest joined probability will be chosen as the n–best. This approach is however purely stochastic and lacks linguistic motivation.

Given the description above, a Speech Recogniser (`SRer`) can be formally described as a function –f– defined in terms of the reference lexicon acoustic patterns (`APat`) and the language model (`LM`) extracted from a training corpus:

$$SRer = f(APat, LM)$$

A Speech Recognition System (`SRSys`) is a function –g– which uses the recogniser (`SRer`) and an acoustic input (`AInput`) to generate the process representation in the form of a word lattice (`WL`):

$$\begin{aligned} SRSys &= g(SRer, AInput) \rightarrow WL \\ &= g(f(APat, LM), AInput) \rightarrow WL \end{aligned}$$

3.3.2 Semantic Interpretation

In what this project is concerned, the semantic interpreter can be described as a unification–based CFG parser (`UPer`), which can be described as a function –l– defined in terms of a Context Free Grammar (`CFGrammar`) and a set of Unification Rules (`URules`):

$$UPer = l(CFGrammar, URules)$$

The unification–based CFG parsing system (`UPSys`) can be described as a function –m– which uses the parser (`UPer`) and an input sentence (`S`) to output a semantically interpreted sentence (`SISent`):

$$\begin{aligned} UPSys &= m(UPer, S) \rightarrow SISent \\ &= m(l(CFGrammar, URules), S) \rightarrow SISent \end{aligned}$$

3.3.3 Dialogue Management

On the same lines of the functional definitions above, the Dialogue Manager (`DMer`) can be described as a function –r– defined in terms of the Dialogue Specifications (`DSpec`):

$$DMer = r(DSpec)$$

The Dialogue Management System (`DMSys`) can be defined as a function –s– which will use the dialogue manager (`DMer`), the semantically interpreted sentence (`SISent`) and the dialogue history (`DHis`) to generate a set of dialogue actions (`DAct`) and expectations (`Expt`):

$$\begin{aligned} DMSys &= s(DMer, SISent, DHis) \rightarrow DAct\&Expt \\ &= s(r(DSpec), SISent, DHis) \rightarrow DAct\&Exp \end{aligned}$$

3.4 Problems and Solutions

3.4.1 Problem 1: WL vs. S

The first problem we encounter is the interface between the Speech Recognition and Parsing Systems: the Speech Recognition System outputs a word-lattice (WL), whereas the input required by the Parsing System is a sentence or one-dimensional string of words.

N-best algorithms are the most commonly used solution to this problem. However, they lack linguistic motivation and generate semantically and grammatically incorrect strings.

Given the enormous number of possible sequences contained in the WL, using all the candidate combinations as input for the parsing system is not a viable solution.

3.4.2 Solution: Bidirectional Parsing of Word-Lattices

We propose a modified parsing algorithm that will be able to process the information contained in the word-lattice. Since using the full WL itself as input is not a computationally viable solution, we need to translate the information contained in the word-lattice into a more friendly and parsable format: DAG (Directed Acyclic Graphs) structures. The use of DAGs also implies the need for an additional linking process between the speech recognition and the parsing systems to translate the lattice structure generated by the `SRSys` into a DAG structure acceptable by the `UPSys`.

Although the advantages of this integrationist approach may be obvious, this task is by no means trivial. It implies the design of a new formal parsing model that must be robust as well as efficient enough to handle the vast amount of information contained in the word-lattice.

3.4.3 Problem 2: Expectations in the Unification-based Parsing System

The second problem is the integration of the Semantic Interpretation and Dialogue Management Systems. The former has no contextual knowledge of the dialogue in process; in other words, it only uses the utterance being processed as its information source.

3.4.4 Solution: Expectation-Directed Parsing and Semantic Interpretation

We propose the use of the dialogue expectations as an additional source of information during the semantic interpretation. This information will lead the Semantic Interpretation System through the analysis and help disambiguate troublesome constructions. Moreover, this mechanism is not destructive, unlike the use of sub-grammars, which ignores analyses not considered in themselves.

As before, the main challenge we face is the design and implementation of a new parsing model that will include dialogue expectations as an additional and valuable source of information during the semantic interpretation process. The fact that both systems use the same kind of semantic information will facilitate the communication process.

3.4.5 New Parsing Model

We therefore propose a new function –m'– for the unification–based parsing system:

$$\begin{aligned} UPSys &= m'(UPer, WL, Expt) \rightarrow SISent \\ &= m'(l(CFGrammar, URules), WL, Expt) \rightarrow SISent \end{aligned}$$

Where the parsing will be executed in two phases:

1. Preprocessing of WL into DAG structures: The PP-DAG Algorithm (Figure 3.1)
2. Processing of DAG structures to generate a SISent: The SCP Algorithm (Figure 3.2)

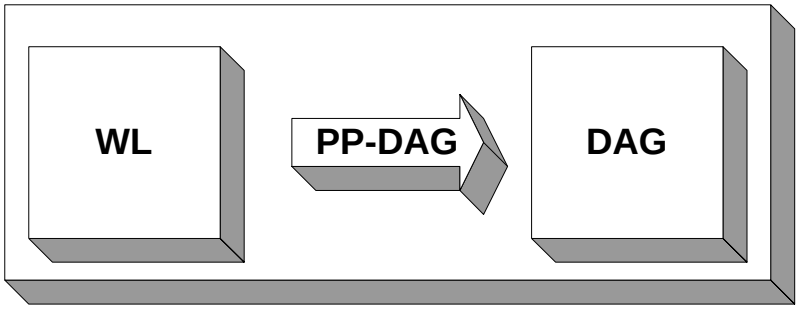

Figure 3.1: Preprocessing of WL into DAG structures

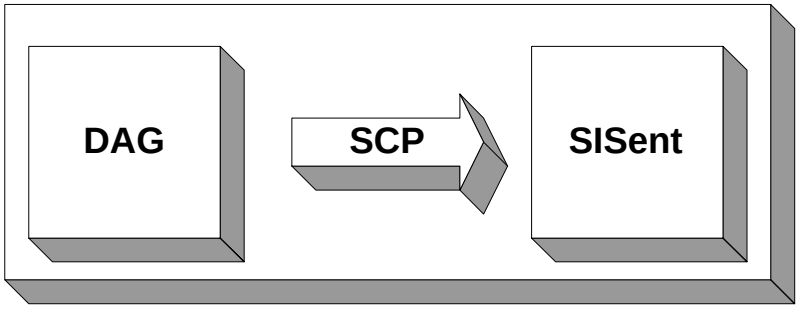

Figure 3.2: Processing of DAG structures to generate a SISent

One of the most striking and innovative ideas in this approach is the fact that the parsing will take place within the `SRSys`, unlike in any other parsing-based approach we have seen, where linguistic parsing is usually used in a post-recognition phase.

Chapter 4

Summary and Conclusions

This deliverable has provided some preliminary results to show that knowledge from sources traditionally kept separate from the speech recognition module in a dialogue system can be used to increase speech recognition accuracy.

Chapter 2 used Information State information as a way to choose between recognition hypotheses. By careful examination of the data, the ways in which the IS could contribute to improved recognition were identified, and one particular example was implemented and showed the expected improvement. Work continues to implement the other identified potential improvements.

Chapters 3 described the University of Seville's approach to integrating semantic and dialogue state information into the speech recogniser. In this approach, parsing and interpretation are done directly on a recogniser word lattice which has been converted into a more compact directed acyclic graph.

Bibliography

[Aho & Ullman 1972] Aho, A. V. & J. D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling. Vol. I: Parsing.* Englewood Cliffs, N.J.: Prentice Hall.

[Danieli & Gerbino, 1995] M. Danieli and E. Gerbino. 1995. Metrics for evaluating dialogue strategies in a spoken language system. *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, 1995, pp. 34–39

[Chelba & Jelinek, 1999] Chelba, C., Jelinek, F. 1999. Recognition performance of a structured language model. *Proceedings of Eurospeech*, 1999, pp. 1567–1570

[Drobot 1989] Drobot, V. 1989. *Formal Languages and Automata Theory.* Rockville, MD: Computer Science Press.

[Khudanpur & Wu, 1999] Khudanpur, S., Wu, J. 1999. A Maximum Entropy Language Model to Integrate N-Grams and Topic Dependencies for Conversational Speech Recognition. *Proceedings of ICASSP'99*, pp. 553–556

[Knight *et al*, 2001] Sylvia Knight, Genevieve Gorrell, Manny Rayner, David Milward, Rob Koeling, and Ian Lewin. 2001. Comparing grammar-based and robust approaches to speech understanding: a case study. *Proceedings of Eurospeech*, 2001.

[Koeling,2002] Rob Koeling. 2002. Dialogue-Based Disambiguation: Using Dialogue Status to Improve Speech Understanding. PhD Thesis, University of Groningen, 2002.

[Milward, 2002] David Milward. 2002. Exploiting the advantages of task and linguistically orientated dialogue management. Siridus Deliverable D4.4, 2002.

[Quesada, 1997a] Quesada, J. F. 1997. *El algoritmo SCP de anlisis sintctico mediante propagacin de restricciones.* PhD dissertation. University of Seville.

[Quesada, 1997b] Quesada, J. F. 1997. A General, Sound and Efficient Natural Language Parsing Algorithm based on Syntactic Constraints Propagation. *Proceedings of the VII Conference AEPIA'97*, pp. 775–786.

[Quesada, 1998a] Quesada, J. F. 1998. Bidirectional and Event-Driven Parsing with Multi-Virtual Trees. In Martn-Vide, ed. *Mathematical and Computational Analysis of Natural Language*, John Benjamins, pp. 253–265.

[Quesada, 1998b] Quesada, J. F. 1998. Lexical Object Theory: Specification Level. *Grammars*, 1(1), 57–84.

[Quesada, 1999] Quesada, J. F. 1999. Overparsing. In Martn-Vide, ed. *Issues in Mathematical Linguistics*, John Benjamins, pp. 165–182.

[Van Noord *et al*, 1999] Van Noord, G., Bouma, G. Koeling, R. Nederhof, M.J. 1999. Robust Grammatical Analysis for Spoken Dialogue Systems. *Journal of Natural Language Engineering*, **5(1)**, pp. 45-93.

[Walker *et al*, 1997] Marilyn Walker, Diane Litman, Candace Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics , ACL 97*, 1997.

[Walker *et al*, 2000] Marilyn A. Walker, Lynette Hirschman, and John Aberdeen. 2000. Evaluation for darpa communicator spoken dialogue systems. *Proceedings of the Language Resources and Evaluation Conference, LREC*, 2000.