# Prosodic Cues for Information Structure

Massimo Poesio
James Hieronymus
Stephen Isard
Robin Cooper
Johan Bos
Helen Wright
Staffan Larsson

Distribution: PUBLIC


Task Oriented Instructional Dialogue
LE4-8314

Deliverable D4.2

May 2000

LE4-8314 TRINDI

Task Oriented Instructional Dialogue

**Gothenburg University**
Department of Linguistics

**University of Edinburgh**
Centre for Cognitive Science and Language Technology Group, Human Communication Research Centre

**Universität des Saarlandes**
Department of Computational Linguistics

**SRI Cambridge**

**Xerox Research Centre Europe**

For copies of reports, updates on project activities and other TRINDI-related information, contact:

The TRINDI Project Administrator
Department of Linguistics
Göteborg University
Box 200
S-405 30 Gothenburg, Sweden
trindi@ling.gu.se

Copies of reports and other material can also be accessed from the project's homepage, http://www.ling.gu.se/research/projects/trindi.

# Contents

# Chapter 1

# Introduction

Much evidence points to a correlation between prosodic information and, on the one hand, the focus-ground articulation of the content of single dialogue moves; on the other hand, the organization of these moves in larger games. In this deliverable we present preliminary work concerning the second of these points in speech recognition and both points for speech synthesis.

In previous work, Poesio and Mikheev (1998), we found evidence that information about the organization of moves in games helps move prediction, when working with transcripts. In the first section, we evaluate the contribution of prosodic information in recognizing game structure. We also extend the previous results in two other senses. First of all, we train and test our models of move prediction directly over the speech signal, rather than over transcriptions and annotations. Secondly, we use our own predictions of game structure, rather than what had been previously annotated.

For question generation it is clear that the exact question understood by the user depends on the focus words in the utterance, and how they are marked prosodically. By analyzing the information state, it is possible to determine what the system wants to find out semantically. Given this information and an analysis of the shared beliefs it is possible to generate a question which is clear to the user. This clear question then will elicit a short and precise answer containing the information needed by the system. It should also be possible to analyze the focus structure in each user utterance, but this will be left for future work. We present priminary results of placing focus in questions produced by information structure analysis in the GoDis and MIDAS systems.

# Chapter 2

# Automatically predicting dialogue structure using prosodic features

## 2.1 Introduction

Dialogue act identification is an important task for a dialogue system. It is essential to know if the response to a system's question is an answer or an objection. In addition, it is important to establish the extent to which the user has established a conversational goal so that the dialogue system can update its knowledge base and continue the conversation in the appropriate manner. The goal of the work reported in this paper is to test whether using hierarchical information about dialogue structure leads to improved performance in dialogue act recognition.

As in previous work, Taylor *et al.* (1998), we integrate dialogue act recognition with word recognition, in the sense that word probabilities are computed by language models specific to different dialogue acts, and dialogue act likelihoods take word probabilities into account. The hypotheses produced by the integrated system are of the form "Yes-no query consisting of 'Is it' " or "Reply consisting of 'It is' ", where, crucially, the hypothesised word string for the utterance viewed as a question need not be the same as the hypothesised word string for the same utterance viewed as a reply. As a result, the a priori most likely dialogue act can potentially be rejected on the basis of word recognition and the phonetically most likely word string can be rejected on the basis of dialogue act considerations. The viterbi architecture which achieves this integration is described in section 2.3. Previous studies have shown that a reduction of word error rate is obtainable by integrating dialogue act recognition into their systems, Taylor *et al.* (1998); Shriberg *et al.* (1998).

The architecture described in section 2.3 also permits us to introduce intonational information in a natural way. Dialogue acts correlate not just with the words that are spoken, but with how they are spoken, in particular with prosody. For example, in our data the utterance "okay" is often realised with a rising intonation if it is a *checking* dialogue act and a falling intonation if it is an *acknowledgement*. Our architecture weights the likelihoods of dialogue

act types for a given utterance according to their probability of occurrence with the observed intonation contour calculated using a statistical intonation model.

As well as modelling the word sequences and the intonation of various dialogue acts, our system uses a dialogue model that captures regularities in a sequence of dialogue acts. For example, a *query* followed by a *reply* followed by an *acknowledgement* is more likely than three replies in a row. The theory of dialogue that we adopt is derived from the theory of *Conversational Games* used to annotate the Map Task corpus, Power (1979); Carletta *et al.* (1997). According to this theory, conversations consist of a series of GAMES, each of which involves an INITIATING MOVE (such as an *instruction* or a *query*) followed by either a RESPONSE MOVE (such as an *acknowledgement* or a *reply*) or possibly an embedded game (e.g., a *query* may be followed by a *clarification* subdialogue).

Experiments reported in Poesio and Mikheev (1998) used the annotated Glasgow version of the Map Task corpus to compare the ability of two types of dialogue models to predict move types. The first type takes the hierarchical structure of Conversational Game Theory into account; the second simply models the sequence of moves ignoring game structure, as in the models used in Nagata and Morimoto (1994); Reithinger and Klesen (1997); Taylor *et al.* (1998). Poesio and Mikheev found that having perfect knowledge of a move's position in a game and of the type of game leads to a 30% reduction in the error rate of move prediction.

The goal of the experiments described below, was to compare various dialogue models in terms of their ability to predict the move type of an utterance whose game information is automatically derived. We find that taking into account the position of an utterance in a game significantly improves the ability of the system to predict move type, even when this information has to be automatically extracted from the input.

We also look at whether classifying utterances using game information provides a better correlation with observed intonation patterns. For example, a *ready* move at the start of a game may be more emphatic than one in the middle of a game. Finally, we test whether knowing the game position a move gives us extra information about word sequence regularities. For instance, a *ready* move at the start of a game may contain a larger vocabulary than *ready* moves in the rest of the game, as these just tend to consist of "okay".

We first discuss the type of data used and the general architecture of our system. We then describe each of the statistical models in turn (dialogue, intonation and language models) and how game information can make these models more effective. Finally, we present move recognition results and discuss further possible developments.

## 2.2 Data

The experiments reported here use a subset of the DCIEM Maptask corpus, Bard *et al.* (1995). This is a corpus of spontaneous goal-directed dialogue speech collected from Canadian speakers. This Maptask corpus was chosen as it is readily available, easy to analyse, has a limited vocabulary and structured speaker roles. Each conversation has two participants each

with different roles called the *giver* and *follower*. Generally the *giver* is giving instructions and guiding the *follower* through the route on the map. Due to the different nature of the roles, each participant has a different distribution of moves.

As described above the corpus has been analysed using the game-move theory modified for Maptask dialogues. Game and move information was hand-labelled for a set of 25 dialogues which we divide into a training set of 20 dialogues (3726 utterances) and a test set of 5 dialogues (1061 utterances). None of the test set speakers are in the training set, i.e. the system is speaker independent.

## 2.3 System architecture

As discussed in the introduction, our system performs move recognition using three types of statistical models: intonation model (IM), dialogue models (DM) and language models (LM) in addition to the output of the speech recogniser.

Although there is a correlation between intonation contour types and move types, there is not a unique mapping, any more than there is for syntactic types or dialogue contexts. For example, *align* move types are realised with both rising and falling boundary tones, possibly reflecting the level of the speaker's confidence. Wright (1999) describes methods for training stochastic models that assign a likelihood for each move type given the current pitch contour. These likelihoods are combined with the outputs of the other components to produce an overall best guess. The use of stochastic models is only successful if each move type has a different distribution of intonation features.

In Taylor *et al.* (1998), a separate language model is trained for each move type, resulting in twelve language models for the original move set. The speech recogniser is effectively run several times in parallel, using each of the move specific LMs. Language model prior probabilities are combined with word recognition probabilities to produce likelihoods for word strings according to the various language models. For example, if the recogniser assigns a high probability to a hypothesis of the word "yes" spanning the whole utterance, the *reply-y* LM will produce a high score overall, because the probability of "yes" as a *reply-y* is also high. However, the *reply-n* LM will produce a lower score because the high recognition score for "yes" will be multiplied by a low probability of occurrence in that LM.

Finally, regularities about move types are captured by a statistical dialogue model. The dialogue models we tested use dialogue information such as the previous move type, the position of a move in a game, the type of a game, and the identities of the speakers to predict the current move type.

A viterbi search finds the most likely sequence of moves together with the words used to perform them. This process searches through all the possible move sequences, given the likelihoods from the intonation models and the language models. The probability of a sequence of moves is the product of the *transition probability*, given by the dialogue model, and the *state probability*, which is a combination of the likelihoods from the prosodic and language

models. These likelihoods are weighted and summed in the log domain using the following equation, where $M\star$ is the most likely move type sequence:

$$M^{\star} \;=\; \underset{M}{\operatorname{argmax}} \left\{ w_D L^D \;+\; \sum_{i=1}^{N_U} \left( w_S L_i^S \;+\; w_I L_i^I \right) \right\} \tag{2.1}$$

where $L^D$ is the log likelihood from the dialogue model; $L_i^S$ and $L_i^I$ are the log likelihoods for utterance $i$ from the speech recogniser and intonation model respectively; $w_D$, $w_S$ and $w_I$ are the weights for the three terms. This method is illustrated in figure 2.1, taken from Taylor *et al.* (1998).

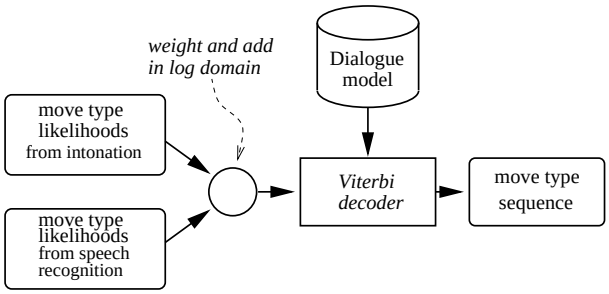


Figure 2.1: Finding the best move sequence

The weights are found using a held out data set, as proposed by King (1998). The intonation model and recogniser weights are systematically varied, while keeping the dialogue model at a fixed weight, until the optimal move recognition rate is achieved.

The result of the viterbi search is a sequence of the most likely move types for each utterance, together with the word sequence recognised by the most likely move type-specific language model. This word sequence is not irrevocably chosen before intonation and dialogue information are taken into account.

The results we present in this paper show that an improvement over previous attempts at move recognition is achieved by automatically recognising the position of the utterance in a game as well as the move type. For comparison purposes, the accuracy of the system is calculated in terms of the percentage of utterances whose move type is correctly classified.

For evaluating our language, dialogue and intonation models on their own, in isolation from the rest of the system, we use the "value added" type measure of perplexity reduction. Perplexity is an information theoretic measure which rates a task as being "as hard as choosing among n equally likely possibilities" (see Rabiner and Juang (1994), p 449). The contribution that we are hoping for from each of our models is to reduce the perplexity of move recognition by as much as possible.

| Move | Start | Middle | End | Total |
|---|---|---|---|---|
| *acknowledge* | 0 | 409 | 510 | 919 |
| *align* | 95 | 22 | 4 | 121 |
| *check* | 185 | 51 | 9 | 245 |
| *clarify* | 0 | 66 | 25 | 91 |
| *explain* | 192 | 96 | 43 | 331 |
| *instruct* | 192 | 381 | 43 | 606 |
| *query-w* | 78 | 17 | 2 | 97 |
| *query-yn* | 237 | 93 | 4 | 334 |
| *ready* | 271 | 70 | 5 | 346 |
| *reply-n* | 0 | 82 | 25 | 107 |
| *reply-w* | 0 | 116 | 29 | 145 |
| *reply-y* | 0 | 201 | 183 | 384 |
| total | 1240 | 1604 | 882 | 3726 |

Table 2.1: Move frequencies with respect to game position

| Predictor | Symbol |
|---|---|
| Move_position type of current move | $m\_p_i$ |
| Identity of speaker of current move | $s_i$ |
| Identity of speaker of previous move | $s_{i-1}$ |
| Move_position of previous utterance | $m\_p_{i-1}$ |
| Move_position of other speaker's previous utterance | $m\_p_{other}$ |

Table 2.2: Notation of N-gram predictors

## 2.4 Using Game Position in Dialogue Modelling

In studies such as Nagata and Morimoto (1994); Reithinger and Klesen (1997); Taylor *et al.* (1998); Shriberg *et al.* (1998); King (1998) dialogue is assumed to have a flat structure, and the current dialogue act or move type is predicted on the basis of the previous utterance type only (possibly taking into account information about the current and previous speaker as well). In this section, we show that dialogue models that encode information about the position of a move in a game can reduce the perplexity of the test set.

In order for this dialogue model to give good results, there must be a distinctive distribution of move types with respect to their game position. Table 2.1 gives the frequencies of the different moves in different game positions for the training set. From this table, one can see that there are clear patterns of move distributions across game positions. These regularities should be picked up by the dialogue model. For example, an obvious pattern is that initiating moves, with the exception of *instruct*, occur most frequently at the start of games. Most *ready* moves are game initial. Replies are quite evenly distributed across middle and end positions. All replies, with the exception of *acknowledge*, have a higher frequency of middle moves than game final moves.

Table 2.2 gives the types of predictors we used in training N-grams for dialogue modelling,

| Model | Predictors | Perplexity |
|---|---|---|
| A | unigram | 18.7 |
| B | $m\_p_{i-1}$ | 9.8 |
| C | $m\_p_{i-1}$, $s_i$, $s_{i-1}$ | 8.55 |
| D | $m\_p_{other}$, $s_i$, $s_{i-1}$ | 7.6 |

Table 2.3: Perplexity results for the different dialogue models predicting move_position categories

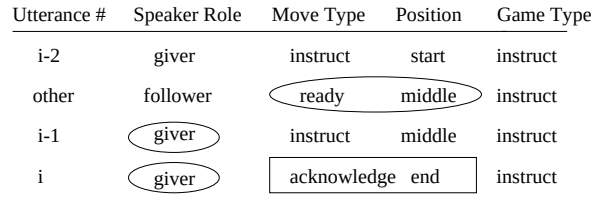


Figure 2.2: Illustration of the predictors (circled) used in Model D for predicting the move and position of the current utterance (boxed)

(see Jelinek and Mercer (1980)). Several combinations of these predictors were used for determining the move_position of an utterance ($m\_p_i$). The test set perplexities of the different combinations are given in table 2.3. The lower the perplexity, the more predictive the dialogue model. As shown in previous dialogue modelling experiments (King (1998); Taylor *et al.* (1998); Chu-Carroll (1998); Poesio and Mikheev (1998)), speaker identities are good predictors of moves in task-oriented conversations. This is the case when the different roles played by the conversational participants (*giver* and *follower* in the Map Task) lead to different distributions of move types. The 4-gram that reduces the perplexity the most (Model D) uses the move_position type of the other speaker's previous move ($m\_p_{other}$) and the current and previous speaker type. This model is illustrated in figure 2.2.

### 2.4.1 Modifying the Move_position Utterance Type Set

One can see from table 2.1 that some combinations of move and position are infrequent, such as replies at the start of games and queries at the end of games. This results in sparse data problems, especially for the language and intonation models described below. Therefore, a modified set of move_position utterance types was derived by combining some of the less frequent categories[1]. This new set is referred to as MOVE_POSITION SET 2 and contains 19 categories.

A complete list of move_position set 2 types is given in table 2.4. The *end* and *middle* moves are combined for the following move types: *instruct*, *query-w*, *query-yn* and *ready*. This is motivated by the lack of game final utterances of these types. The start and middle categories are merged for the following move types: *reply-n, reply-y* and *acknowledge*. This is motivated by the lack of game initial utterances of these move types.

[1] A number of preliminary experiments were conducted that examined language model perplexities and intonation similarities to find the classification scheme that had the most potential.

| Start | Middle | End |
|---|---|---|
| acknowledge_middle | | acknowledge_end |
| align | | |
| check | | |
| clarify | | |
| explain | | |
| instruct_start | instruct_middle | |
| query-w_start | query-w_middle | |
| query-yn_start | query-yn_middle | |
| ready_start | ready_middle | |
| reply-n_middle | | reply-n_end |
| reply-w | | |
| reply-y_middle | | reply-y_end |

Table 2.4: Move frequencies with respect to game position

| Model | Predictors | Perplexity |
|---|---|---|
| A | unigram | 14 |
| B | $m\_p_{i-1}$ | 8.3 |
| C | $m\_p2_{i-1}$, $s_i$, $s_{i-1}$ | 6.9 |
| D | $m\_p2_{other}$, $s_i$, $s_{i-1}$ | 4.5 |

Table 2.5: move_position set 2 perplexity results for the different dialogue models

The following moves are not distinguished by their game position: *align, check, clarify, reply-w*, and *explain*. These moves have a longer, more varied syntax. Language modelling experiments described below show that it is beneficial to use one category for these utterances as this allows more data for training the models. Shorter, less varied utterance types such as acknowledgements and replies need less data for training, for example positive replies usually contain one of a small set of words "yes, yep, yeah, etc.". The utterance type recognition baseline is lower than the original move_position set; the most frequent move is *acknowledge_end*, which makes up 13% of the data.

### 2.4.2 Dialogue Models for move_position set 2

A number of dialogue models were developed to predict the move_position set 2 utterance types. The perplexity of the test set using these models is given in table 2.5. Again, the best perplexity result (4.5) is achieved by using the other person's previous utterance type ($m\_p2_{other}$) and speaker identities (model D). This new dialogue model D was used in conjunction with the intonation model and language models in the experiments described below.

## 2.5 Game Position and Intonation Modelling

Previous studies have shown that intonation can be indicative of the position of a move in a game. For example, Nakajima and Allen (1993) show that average F0 at the start and end of an utterance varies depending on whether the utterance is continuing or introducing a new topic. This suggests that moves of the same type may differ in intonation depending on their position in the game. If an utterance is game initial it may be introducing a new goal or topic and have a slightly higher utterance initial F0 contour. In order to investigate this potential correlation, we trained statistical intonation models to distinguish the combined move and position utterance types.

Wright (1998) describes three methods for modelling intonation using stochastic intonation models: hidden Markov models, classification and regression trees (CART), and neural networks. As she concludes that CART trees are slightly more effective than the other two systems, we adopted this method in our experiments here. Forty-five suprasegmental and durational features were used to construct tree structured classification rules, using the CART training algorithm, (see Breiman *et al.* (1984)). The tree can be examined to determine which features are the most discriminatory in move classification.

The output of the classification tree is the probability of the move ($M$) given the observed intonation features ($I$), i.e. the posterior probability $P(M|I)$. However, in order to be able to use the output of the CART model in the system described in section 2.3, we need to derive the likelihood $P(I|M)$ rather than the posterior probability. This can be calculated in two ways. Firstly, one can train the CART tree on equal numbers of each utterance type. A second method is to divide the posterior probability by the prior probability $P(M)$. These two methods produce similar results.

### 2.5.1 Intonation Features

The suprasegmental features are automatically extracted from the speech signal and used to train the classification tree. For each move the last three accents (if present) are automatically detected using a method described in Taylor (2000). This method identifies accents (a) and rising or falling boundary tones (rb/fb). In order to determine the type of the accents, they are automatically parameterised into four continuous *tilt parameters*: start F0, F0 amplitude, accent duration and *tilt*. Tilt is a figure between -1 and 1 and describes the relative amount of rise and fall of the pitch contour for an accent (see Taylor (2000)).

A set of more global features based on the study by Shriberg *et al.* (1998) is also extracted. These are prosodic features based on F0 (e.g., max F0, F0 mean and standard deviation), root mean squared (RMS) energy (e.g., energy mean and standard deviation) and duration (e.g., number of frames in utterance, number of frames of F0). These features are calculated for the whole utterance, for example, the standard deviation of the F0 represents pitch range. The least-squares regression line of the F0 contour is also calculated. This captures intonation features such as declination over the whole utterance. In addition, the above-mentioned features are calculated for the final and penultimate part of the intonation contour which is

| Feature Type | Usage (%) |
|---|---|
| Duration | 47 |
| F0 | 41 |
| RMS Energy | 12 |

Table 2.6: Discriminatory features and type usage in move classification

often indicative of utterance type. For example, the least square error for F0 in the final part of the contour is indicative of the type of boundary tone. Other features are calculated by comparing feature values for the last two regions and the whole utterance (e.g., ratio of mean F0 in the end and penultimate regions, difference between mean RMS energy in the end and penultimate regions). A comprehensive list of these features is given in Appendix A.

It is useful to know which features are the most discriminatory in the classification of the moves. As the tree is reasonably large with 30 leaves, interpretation is not straightforward. For simplicity, we group the features into three general categories: duration, F0 and energy. Table 2.6 gives the *feature usage frequency* for these groups of features. This measure is the number of times a feature is used in the classification of data points of the training set. It reflects the position in the classification tree as the higher the feature is in the tree, the more times it will be queried. The measure is normalised to sum to 100% for the whole tree.

Different move types by their nature vary in length, so it is not surprising that duration is highly discriminatory in classifying utterance types. For example, *ready, acknowledge, reply-yes, reply-n* and *align* are distinguished from the other moves by the top node which queries a duration feature. This duration feature, regr_num_frames, is the number of frames used to compute the F0 regression line for a smoothed F0 contour over the whole utterance. This is comparable to the study reported in Shriberg *et al.* (1998), where durational features were used 55% of the time and the most queried feature was also regr_num_frames. This feature is an accurate measure of actual speech duration as it excludes pauses and silences.

The F0 features that are used frequently in the tree are F0 mean in the end region, maximum F0 and tilt value of the last accent. For example, in one part of the tree *align* moves are distinguished from *instruct* moves by having a higher F0 mean for the end region which may indicate boundary tone type.

### 2.5.2 Classification Results using the Intonation Model

A classification tree was trained on the features mentioned above to distinguish between the 19 categories in table 2.4. The results of these recognition experiments are given in table 2.7. Using the intonation model alone achieves a recognition rate of 30%, which is significantly higher than the baseline (13%). Dialogue model D has a recognition rate of 25%. Combining the intonation and dialogue models yields 37% correct. This is a 12% increase over the dialogue model alone.

The effectiveness of intonation models is very hard to judge. However, as the recognition

| | Original moves % | move_position set 2 % |
|---|---|---|
| Baseline | 24 | 13 |
| Intonation Model | 45 | 30 |
| 4-gram | 37 | 25 |
| 4-gram & Intonation | 47 | 37 |

Table 2.7: % of utterances correctly recognised for move and move_position set 2 utterance types using Model D and intonation models

results are well above the baseline, one can assume that they incorporate some of the distinguishing characteristics of the different utterance types.

## 2.6 Game Position and Language Modelling

Taylor *et al.* (1998) trained separate language models for utterances of each move type, thus capturing the lexical characteristics of each type. They show that by using these move-specific language models they can reduce the perplexity of word sequences in comparison with a general language model[2]. These language models are used to determine the likelihood of an utterance belonging to one type or another. As discussed in section 2.3, this is achieved by running the recogniser 12 times using each of the language models, and then choosing the move type whose associated language model produces the highest probability.

Language models are smoothed with a general model. This compensates for sparse data while still capturing the characteristics of the specific move types. For each move the perplexities of the general, move specific and smoothed models are compared and the lowest one is chosen. This result is known as the *best choice* result.

Similar language modelling experiments were run for move_position and move_position set 2. Our language models were trained using the CMU Language Modelling toolkit (see Rosenfeld and Clarkson (1997)). Similar word sequence perplexity results were obtained for the best choice language modelling experiments. Using the original move type language models yields a perplexity of 23.8[3] whereas the move_position set 2 yields 23.9. This is promising given that the second set contains more moves and therefore there is less data to train the individual models. Using a general language model yields a higher perplexity result of 27.6.

## 2.7 Move Recognition Results

As discussed above, the method for move recognition presented in this paper involves two stages. First, we automatically determine the likelihood of each move_position set 2 utterance

[2]Higher predictability of words is reflected in a lower perplexity.

[3]This result is not comparable with that in Taylor *et al.* (1998) where a larger training set of 40 dialogues was used.

| | Models Used for Move Recognition | % Correct for Move Recognition | % Correct for Move Recognition collapsing m_p2 |
|---|---|---|---|
| A | Baseline | 24 | 24 |
| B | DM only | 37 | 37 |
| C | Recogniser output and LM | 40 | 45 |
| D | Recogniser output and LM and DM | 57 | 64 |
| E | IM | 42 | 43 |
| F | IM and DM | 47 | 50 |
| G | DM, IM, recogniser output and LM | **64** | **66** |

Table 2.8: Move detection accuracy using various information sources

type. The classification of utterances in terms of this utterance type scheme is 49%, with a baseline figure of 13%, which is the classification accuracy if *acknowledge_end* is chosen 100% of the time.

The move_position set 2 utterance types are then collapsed to obtain the likelihood of each move type. Table 2.8 gives the results for move classification after the move_position set 2 utterance labels have been collapsed. With the exception of experiment B (in which only the dialogue model is used), all the recognition results are increased using the new utterance types that encode the position in the game. The system as a whole increases its accuracy from 64% to 66%. Although this increase is small, it is found to be significant by a Sign test (see Siegel and Castellan (1988)) ($p < 0.01, d.f. = 1060$)[4].

The confusion matrix of moves correctly recognised by the whole system is given in the matrix in table 2.9. The final column in this table gives the percentage of moves correctly recognised by the system that does not use position[5]. There are several noticeable differences between the two sets of results. Firstly, using position leads to fewer *acknowledges* being misrecognised as *ready* moves, as these rarely occur in the same game position. This improvement in *acknowledge* recognition also accounts for most of the significant improvement in the experiments. Carletta *et al.* (1997) show that mistaking *acknowledges* for *ready* is also a common recognition mistake made by human labellers. Other confusions that humans make include misrecognising *query-yn* as *checks*; *ready* as *reply-ys*; and *clarifies* as *instructs*. These confu-

[4]The Sign test examines the utterances which are classified differently by the two systems. A positive sign is given when the new system is correct and a negative sign when the original system is correct. The null hypothesis tested is that there will be more negative signs than positive ones, positing that the original system is superior to the new system.

[5]For a complete matrix see King (1998) and Wright (1999).

| | acknowledge | align | check | clarify | explain | instruct | query-w | query-yn | ready | reply-n | reply-w | reply-y | Correct % | Original % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| acknowledge | **232** | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 13 | 0 | 1 | 10 | **90** | 80 |
| align | 9 | **6** | 2 | 1 | 1 | 8 | 1 | 6 | 21 | 0 | 1 | 0 | **11** | 3 |
| check | 7 | 1 | **30** | 0 | 4 | 2 | 1 | 16 | 1 | 1 | 1 | 3 | **45** | 41 |
| clarify | 0 | 1 | 0 | **5** | 0 | 14 | 1 | 0 | 0 | 0 | 3 | 0 | 21 | **25** |
| explain | 8 | 2 | 6 | 1 | **53** | 15 | 2 | 7 | 1 | 3 | 5 | 2 | **51** | 37 |
| instruct | 1 | 1 | 3 | 1 | 9 | **171** | 5 | 3 | 2 | 0 | 1 | 3 | 86 | **88** |
| query-w | 3 | 0 | 1 | 0 | 3 | 1 | **10** | 2 | 0 | 1 | 1 | 2 | **42** | 16 |
| query-yn | 3 | 2 | 11 | 0 | 9 | 5 | 2 | **50** | 1 | 1 | 1 | 1 | 58 | **62** |
| ready | 32 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | **41** | 1 | 0 | 2 | 53 | **62** |
| reply-n | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | **25** | 1 | 0 | **86** | 79 |
| reply-w | 4 | 0 | 1 | 0 | 6 | 6 | 1 | 0 | 0 | 0 | **7** | 0 | **28** | 26 |
| reply-y | 24 | 0 | 1 | 0 | 2 | 3 | 0 | 4 | 0 | 1 | 1 | **72** | 67 | **70** |

Table 2.9: Confusion matrix for move type classification: 66% move recognition accuracy

sions are also observed in the above matrix; however, the confusability of these move types is lower than in the original classification matrix (see Wright (1999) for details).

Another gain that comes from taking position into account is that fewer *explain* moves are recognised as replies. This is due to the fact that *explains* mostly occur game initially, whereas replies are mostly game final. There is a 28% increase in *query-w* recognition as fewer of these move types are confused with *acknowledges*. These improvements are attributable to the dialogue model component as these move types rarely occur in the same game position. On the other hand, the dialogue model confuses more *query-yn* moves with *explains* as the majority of both these move types are game initial.

There is an increase in *ready* moves that are misclassified as *acknowledges* despite the fact that they rarely occur in the same game position. On examination of the separate components, we find that this is due to the fact that the language models have a high weighting and both move types have similar wording, i.e. mostly "okay". The intonation models alone have a higher recognition accuracy for *ready* moves (64%).

Using position does not make much difference in recognising replies. This is because they have a fixed syntax that does not vary much across game position.

## 2.8 Conclusion

We studied the relationship between move type prediction and game structure, in particular the position of an utterance in the game. Move type and game position were predicted simultaneously using three statistical models: dialogue, intonation and language models in

conjunction with the recogniser output. Incorporating hierarchical dialogue information into the system resulted in a statistically significant improvement in move recognition.

One issue with a system such as the one discussed above is that the results are very dependent on the discourse analysis theory adopted. The discussions above have shown how difficult it is to develop models that capture both the syntactic and intonation similarities of utterances. One area of future development would be the automatic clustering of utterances by calculating some measure of distance between vectors of words or intonation features. This may result in more distinctive language and intonation models.

Another approach would be to develop context dependent categories. A study conducted by Hockey *et al.* (1997) indicates that the lexical content of a move can be predicted to a certain extent depending on the previous move. For example, there is a low probability of the word "no" if the move is preceded by an *align* move. One can hypothesise that if this is the case, then the move will be intonationally marked. Other intonationally marked moves may be non-replies preceded by queries. Training models based on move type distinguished by their context may result in sparse data problems. As with all recognition tasks, more data would result in better trained models and improved recognition results.

In conclusion, this study is an extension of previous work and has shown that using higher level game information can significantly improve the accuracy of the system in the classification of utterances into different dialogue act types.

# Chapter 3

# Using prosodic features to improve dialogue understanding

## 3.1 Introduction

This chapter describes a method of synthesizing the speech output of the TRINDI dialogue system using prosodic features derived from the Information States in the dialogue manager. For questions, the system determines the focus words in the query being constructed by examining the Information State for the dialogue at the present moment. The exact nature of the information being sought is determined from the information state, and then the prosody of the sentence is constructed to elicit the correct impression of the question being asked by the system. This should make it easier for the user to understand the questions which the system asks, and provide the correct information. For example, if the system needs to determine the destination city in a travel dialogue the system can ask the question

WHICH CITY would you like to go to?

or WHERE would you like to go?

The focus on the correct words makes it clear to the user the nature of the question being asked. Often the default focus from a speech synthesizer is simply wrong for the question being asked and this confuses the user. This is because the speech synthesizer is producing a "neutral" reading, and is making safe words the focus.

There have been many studies of intonation in dialogues which relate discourse structure and prosody. Grosz and Sidner (1986) related intentions to dialogue structure. Several studies have shown that prosody is a good marker of discourse structure. Studies of spoken monologues, either read aloud (Bruce (1982), Grosz and Hirschberg (1992), Lehiste (1980), Thorsen (1985), Brubaker (1972) and Sluijter and Terken (1993)) or spoken spontaneously (Swerts and Geluykens (1994), Swerts *et al.* (1994), Swerts (1994), and Swerts (1997)) have shown that information units are marked by variations in speaking rate, pitch range, loudness, duration of

segments and boundary tones. There have also been a few studies on differentiating between statements and questions. This distinction is lessened by the fact that wh questions and yes/no questions do not have rising intonation at the end, but rather have falling intonation like a statement.

The correlation of pitch accent and discourse elements has been studied by Nakatani (1996) for spontaneous narratives. She found that while new information is generally accented and pronouns are unaccented, this is not always the case. Speakers accent new information and old information when the focus is shifting to this item. Pronouns can be accented to maintain backward looking focus person or thing in focus, or to bring a non-focused given referent into focus. This means that listeners expect to have certain parts of a question, for example, be in focus, and marked prosodically. This includes new information, items to be contrasted, and changes of focus. She found that in the naratives, there were levels of discourse which were arranged in a hierarchy. When a digression about one of the people mentioned in the narative is begun, a whole new set of focus persons is in focus, and accented accordingly. When the narative returns to the main subject, it is as if the previous segment was pushed on a stack, and then returns. In this case either an accented pronoun or accented name, signals the topic shift back to the non-digressive one.

This is an important finding for the Trindi system, since conformation questions are brought into the qud and addressed, then popped off to shared beliefs upon completion. This then brings the previous items on qud into focus. By using an accented pronoun to bring the previous dialogue into focus, the system may be able to signal the topic change without repeating the whole previous question. Various strategies have been tested for effectiveness and naturalness.

Recent work by Nakatani and Chu-Carroll (2000) uses this theory to provide accentuation for descriptions of where and when movies are playing in an information giving dialogue system called MIMIC. This system accents all of the information bearing words which appear as attributes in attribute value pairs which supply the movie information. One example of an accent marked up statement from their paper is,

$$*L+H.8* \ Analyze \ *L+H.8* \ This\ is\ playing\ at \ *H*.8 \ Wellmont$$
$$*H.8 \ *L-H\%.8 \ Theatre \ and \ *H*.6 \ Clearview \ *H*.6 \ Screening$$
$$*H*.6 \ *L-H\%.6 \ Zone \ in \ *L+H*.8 \ *L-H\%.8 \ Montclair.$$

Where the symbols represent escape sequences which control the tunes in the ToBI labeling scheme, and the decimals represent strengths of the accent, 1 being the highest. As can be immediately seen the subject which represents the movie title is accented, as are the theatre names and the town name. These are the attribute values for the fields

Movie title:

Theatre Name:

Town Name:

These examples seems somewhat overaccented in casual listening. However, the implementations of the pitch accents are not correct in the present software which we have for the Lucent TTS system. The research system is much more capable of faithfully producing these tones.

A previous system from Hiyakumo *et al.* (1997) at the MIT Media lab used a similar accenting scheme to make news and instructional material more interesting in intonation. They analyze the text into theme (topic) and rheme (comment), and make them with the ToBI tunes L+H* and H* respectively, following the work of Provost and Steedman (1994) on intonation generation. They postulate that each type of constituent has a characteristic ToBI tune, so that analyzing the text to find new themes and rhemes and accenting these appropriately will result in more natural sounding utterances. The special case which they dealt with is the case of synonyms, which would not be accented in first mention, because This example appears in the paper

`The` $*L+H*$ *cloning of an* $*L+H*$ *adult* $*L+H*$ *sheep in* $*L+H*$ *Scotland* $*L+H*$ *seems* $*L-H*$ *likely to* $*H*$ *spark an* $*H*$ *intense* $*H*$ *debate* `about the` $*H*$ *ethics of* $*H*$ *genetic* $*H*$ *engineering* $*H*$ *research in* $*H*$ $*L-L\%$ *humans.*

In casual listening this passage seems overly accented, and leaves the listener breathless. It seems that accenting is something which needs to be done minimally rather than accenting every possibly important word in the discourse. Much of the information in a news cast sentence may be new, for example, but not all of it is accented. Studies by Hieronymus and Williams (1991) showed that an average of 3 accented words occurred in each sentence of a spontanous British English monologue. Beyond this number of accented words the sentence begins to sound un-natural and confusing to the listener.

## 3.2 Assigning Focus from Information States

Information states may be examined to find the best words on which to place the focus for question generation. Some of the considerations behind determining the focus are discussed in Trindi Document D4.1. We will concentrate on question generation here and how the information states in a hypothetical system can be examined to determine the best intonation for the question that the system asks at a particular point in the dialogue. At some point it may be necessary to expand the information within the information state in order to have all the information for generating appropriate questions. In the following section, a simple scheme for exploiting the present GoDis information states is presented. This section discusses the problem in general.

In order to generate precise meanings in synthesized questions it is necessary to know what items have been previously mentioned, whether they were the focus in the previous mention, and where the dialogue is in terms of subdialogues. As previously discussed, given information tends to be unaccented, except when it is brought back into focus after a digression, or if it

has not previously been in focus and needs to be brought into focus in the present turn. This may be done either with explicit mention or by accented pronominal forms.

The previously mentioned words and their accent value must be kept in some sort of history list in order to determine the giveness and focus of words already mentioned. Of course this list would have to include the words recognized by the system too. Since synonyms are also considered as given information, some sort of analysis of the words in the questions to be asked would have to be made to fine tune the question synthesis. Things which appear on the shared beliefs can be considered to have been focus and given words, and to be out of focus at present. For the simple travel dialogues this would include, distination cities, mode of travel, day and time of departure/arrival.

Unfortunately in the present systems there is not an exhaustive list of the items which have been mentioned and their focus state. It would be relatively easy to keep an utterance history, which could include focus, at least for the synthesis.

## 3.3 Predicting intonation from Information States in GoDis

In this section we describe a simple way of predicting intonation for speech synthesis from the information states used in the GoDiS implementation. Our aim here is to use the information already present in our simple information states and to see how far we can get without giving a detailed analysis of intonational phrases. While we believe that a more detailed analysis of both informational and intonational structure could be important for more complex examples, we wish to suggest that simple dialogue systems can be greatly improved by a straightforward linking of text marked with focus intonation to information states which represent a view of the common ground established in the dialogue.

We conducted an experiment by connecting our simple GoDiS system to a standard off the shelf text-to-speech system, IBM's ViaVoice Outloud. Like most TTS systems, ViaVoice Outloud will generate speech from arbitrary unannotated written strings. The basic form of GoDiS produces such strings as responses so the most straightforward way of using ViaVoice Outloud to produce speech for GoDiS is to feed these strings into the TTS system without any additional annotation. If you do this it becomes abundantly clear that the default intonation provided by the TTS system is nearly always inappropriate to what is needed in the dialogue at the particular point where the utterance occurs. We discovered that this was the case for utterances of very simple informational structure, that is, one does not need to look at sophisticated examples of focus constructions in order to find examples where the TTS will make the wrong default predictions about intonation.

We will step through part of a simple dialogue here, indicating the default intonation prediction of ViaVoice Outloud and how an improved intonation can be predicted from the system's information state at that point in the dialogue. The aspect of intonation we are concerned with is the placement of focus accent. At this point we are not interested in the detailed nature of the focus accent or the intonation pattern of the rest of the sentence. This is not to say that better results could not be obtained with increased attention to intonation. At

this point we just wish to show that a marked improvement over the default can be obtained simply by marking where a user perceives the focus accent to occur. We will represent words with perceived focus accent by small capitals here.

If the user begins a dialogue with GoDiS with

`User: Hello`

The system will respond with

`System: What can I do for you?`

ViaVoice Outloud's default intonation for this is

`What can I do for` YOU`?`

which gives the impression that the system is contrasting the user with another user or that the system is surprised that the user is talking to it at all. A better intonation would be

`What can I` DO `for you?`

Given the simple nature of the GoDiS system and the fact that this utterance always occurs at the same point in the conversation, it would be a simple matter to make the system always pronounce this question this way without trying to predict the intonation from the information state. It may well be that this should be assigned as the default intonation of this sentence. However, it is possible to make the connection to the information state. The question being raised concerns the task to be addressed and is represented in the information state as `X^task(X)`. The intonation could be predicted either from the presence of an action on the agenda to raise this question or from the fact that the set of shared beliefs in the information state does not contain a proposition of the from `task(X)`, i.e. no task has yet been specified in the dialogue.

In the experimental dialogue the user continues

`User: A flight please`

and the system responds

`System: What city do you want to go to?`

ViaVoice Outloud's default intonation for this is

`What city do you want to` GO `to?`

which gives the odd impression that there was something else under discussion that you could do to cities other than go to them. A better intonation would be

`What` CITY `do you want to go to?`

One might take this to be the default intonation for questions beginning with a *wh*-phrase consisting of a *wh*-word followed by a noun[1]. Another kind of solution for similar questions is suggested by Nakatani and Chu-Carroll (2000) in the context of a movie theatre information system. The system tries to fill in a form with various values for attributes and they suggest that attributes should always be given focus stress. For example, one of the attributes is *theatre* and another is *movie* so the system is meant to ask questions something like *What* THEATRE *do you want to go to?* and *What* MOVIE *do you want to see?*. This seems to make correct predictions for the particular system they are implementing but it is not so clear that it adapts straightforwardly to the travel domain with which GoDiS is concerned. The problem is that there are two cities involved in these dialogues, the destination and the origin. What corresponds to attributes in form filling are the two questions that need to be raised which in GoDiS are represented by `X^to(X)` and `X^from(X)`. Thus adapting the "attribute focus" approach directly to GoDiS might lead to the incorrect prediction that the preposition should always carry the focus accent. Of course, part of the problem here is that there is a certain arbitrariness about what attributes there are and how they are represented, e.g. whether the attribute is called TO or DESTINATION or CITY-TO might lead the implementer to different conclusions about which words in the sentence should bare focus accent. What seems important from the perspective of the GoDiS domain is that propositions of the form `to(X)` and `from(X)` form contrastive or parallel pairs and the pronunciation of this question depends on the presence or absence of such propositions in the shared beliefs in the information state. Thus at this point in the dialogue neither are present and *city* should carry the focus accent. Things change when we go to the next exchange in the dialogue.

`User: London`

`System: What city do you want to go from?`

Here ViaVoice Outloud's default intonation with focus on *go* makes the system appear not to have understood what the previous part of the conversation is about. Focus on *city*, which was correct in the previous exchange, gives the confusing impression that the system has not understood or even heard the answer to the previous question and is starting again, but with a slightly different question. What is needed here is focus accent on the preposition.

`What city do you want to go` FROM`?`

We can predict this from the information state because the other member of the contrastive pair is present in the shared beliefs of the information state, namely `to(london)`. Note that it is this that predicts the intonation rather than the fact that the system is repeating its previous utterance verbatim except for the different preposition. If the user had said *a flight to Paris* rather than *a flight, please*, the system would have responded with *What city do you want to go from?* directly without having uttered the previous question. With the strategy based on shared beliefs the system would predict focus on *from* which seems correct. [2] It

[1] In the current version of GoDiS this would have to be hand coded on any string of this form as there is no syntactic analysis in the system.

[2] As an illustration of a simple implementation in GoDiS, an argument "focus" is added to `output_form`

is not entirely obvious that previous or recent utterances are not irrelevant, however. If the question was raised several turns after the information about the destination city had been added it is not so clear that focus on the preposition is most appropriate. This problem does not arise in GoDiS at present but in order to deal with it we may have to complicate the information state by including information about when in the dialogue the shared beliefs were added.See Cooper *et al.* (2000). (See also the discussion of parallellism and focus in D4.1.)

## 3.4 Using QUD in assignment and interpretation of focus

Above we have described how focus intonation can be linked to information state in the current implementation of GoDiS. In this section we show how the theory of Questions Under Discussion and their relation to focus intonation (presented in D4.1) could be used for interpretation and generation of focus in GoDiS given some additional capabilities.

### 3.4.1 QUD-based Focus Assignment

In the examples given in this section, we assume that GoDiS uses a cautious grounding strategy, in the sense that user answer is not integrated until system's feedback question is answered. For example, a question is not popped off QUD until the answer has been confirmed. However, we assume a different realisation of feedback questions, where all information gathered so far is included.

In D4.1 we define the concept of Focal Question Presupposition as in (1).

(3.1) **Focal Question Presupposition (FQP):** If an utterance $u$ has narrow focus over $x$, $u$ (focally) presupposes a question $q$ obtained by abstracting $x$ over (the content of) $u$

This concept can be used to formulate a rule for assigning focus to utterances based on Questions Under Discussion:

(3.2) **QUD-based Focus Assignment (QFA):** If there is a question $q$ topmost on QUD, and an utterance $u$ with content $c$ is to be uttered, and $q$ is obtained by

in the lexicon, as well as clauses indicating what information is needed from the information state – this information is supplied by the generation module at the relevant point during the dialogue, since the lexicon has no way of communicating with the information state. The output string *What city do you want to go from?* can be stored in two forms: `What [F city F] do you want to go from?`, as the default, and *What city do you want to go [F from F]?*, which is chosen when the information state contains a user-specified destination city. '[F ... F]' is a focus feature that can be converted to agree with for example ViaVoice Outloud's annotation scheme.

abstracting component $f$ over $c$, then $c$ should focally presuppose $q$ (i.e. focal stress should be put on the part of $u$ that corresponds to $f$).

While not extremely general in scope, the QFA nevertheless makes correct predications in examples like (1) and (2).

(3.3) S1: How do you want to travel?

U1: A flight please

⇒ QUD=⟨{`?X^how(X)`}⟩
**To generate:** `?how(fly)`
**QFA** ⇒ put accent on "flying"

S2: So you're FLYING ?

U2: Yes

S3: Where do you want to to go?

U3: London

⇒ QUD=⟨{`?X^(how(fly) & dest(X))`}⟩
**To generate:** `?dest(london)&how(fly)`
**QFA** ⇒ put accent on "london"

S4: So you're flying to LONDON ?

(3.4) S1: Where do you want to to go?

U1: London

⇒ QUD=⟨{`?X^dest(X)`}⟩
**To generate:** `?dest(london)`
**QFA** ⇒ put accent on "london"

S2: So you're going to [$_{Focus}$ LONDON ]?

U2: Yes

S3: How do you want to travel to london?

U3: A flight please

⇒ QUD=⟨{`?X^(how(X)&dest(london))`}⟩
**To generate:** `?dest(london)&how(fly)`
**QFA** ⇒ put accent on "flying"

S4: So you're [$_{Focus}$ FLYING ] to london ?

In (-1), the focus in S4 is assigned to "London" since there is a question "Where are you flying" (`?X^(how(fly) & dest(X))`) on QUD which can be obtained by abstracting `london` over the content of S4 (`?dest(london) & how(fly)`). However, in (0) the focus is assigned to "flying" since there is a question "How are you traveling to London" (`?X^(how(X) & dest(london))`) on QUD which can be obtained by abstracting `fly` over the content of S4.

### 3.4.2 Interpreting focus to provide helpful answers

In this section we show how GoDiS could use focus information about user utterances to provide helpful answers. Apart from the ability to recognise focus in speech recognition, the examples given here requires GoDiS to be able to handle different airports, as well as direct and indirects flights.

We assume that GoDiS uses the FQuAcc update rule shown in (1) for adding focally presupposed questions to QUD. For motivation and further explanation of the rule, see D4.1.

(3.5) **Focal Question Accommodation (FQuAcc):** When an utterance $u$ occurs which focally presupposes a question $q$ not topmost on QUD, make $q$ topmost on QUD.

(3.6) U1: Are there any DIRECT flights to Gatwick?

**QUD update, FQuAcc**
⇒ QUD=⟨{ `?(direct(yes) & dest(gatwick)),`
`?X^(direct(X) & dest(gatwick))` }⟩

S1: no,

**QUD downdate** ⇒
QUD=⟨{ `?X^(direct(X) & dest(gatwick))` }⟩

S2: but there is a flight to Gatwick via Copenhagen
**QUD downdate** ⇒
QUD=⟨{ }⟩

In (0), GoDiS first pushes the explicit question onto QUD. Then, the FQuAcc rule uses the focus information to extract the presupposed question "Are there any flights (direct or indirect) to Gatwick?". An answer to this question is then given by the system if the direct question received a "no" answer, since it is not answered by simple saying "no" (however, "yes" would have provided an answer to both questions on QUD).

(3.7) U1: are there any direct flights to GATWICK?

**QUD update, FQuAcc**
⇒ QUD=⟨{ `?(direct(yes) & dest(gatwick)),`
`?X^(direct(yes) & dest(X))` }⟩

S2: no,

**QUD downdate**
⇒ QUD=⟨{ `?X^(direct(yes) & dest(X))` }⟩

S3: but there is a direct flight to Heathrow

**QUD downdate**
⇒ QUD=⟨{ }⟩

In (0), the presupposed question could be paraphrased "To which airports (near London) are there direct flights?", and if the direct question is answered by "no", GoDiS answers the presupposed question since it is still on QUD.

## 3.5 Conclusion

Our preliminary conclusion based on the experiment described in section **??** is that quite simple means can be employed to improve the intonational cues given by a dialogue system provided that we have a representation of the dialogue system's information states. Improvement can be obtained in simple dialogues without analysis of syntactic or prosodic structure which potentially makes for simple and quick prototyping of more natural dialogue systems and easier portability. We do not wish to suggest by this, however, that more detailed analysis would not lead to better results.

Also, the examples in section 3.4 indicate some ways that the notion of Questions Under Discussion could be used both in assignment of focus to system utterances and in the integration of user utterances. While these examples would require some additions to the current GoDiS implementation, most notably the ability to detect focus in user utterances, they indicate how the fairly theoretical approach to focus and parallelism presented in D4.1 could be utilized in a dialogue system.

# Chapter 4

# Using Discourse Representation Structures for Concept-to-Speech Generation in MIDAS

## 4.1 Introduction

This chapter presents the concept-to-speech component of the dialogue system MIDAS, and shows how new information in discourse can be used to improve prosodic properties in the synthesized output. First, we discuss the theoretical foundations, by defining given and new information in semantic representations. Second, we describe the different system components of MIDAS. Finally, we discuss the concept-to-speech facilities of MIDAS.

## 4.2 What's New in Discourse Representation Theory

This section discusses how to represent and determine new information in the framework of Discourse Representation Theory ( Kamp and Reyle (1993) ). Little or no work has been devoted to this issue, with the exception of Ivana Kruijff-Korbayova's work which is not discussed in the present version of this paper.

### 4.2.1 Representing New Information

Following standard DRT, A DRS consists out of a universe (a set of discourse referents) and a set of conditions. Conditions can either be basic or complex. Complex conditions are recursively defined as $K \Rightarrow K'$, $K \vee K'$ and $\neg K$ (where K and $K'$ are DRSs). For basic conditions, we have $P(x_1 \ldots x_n)$ for a predicate symbol P with arity $n$ and discourse referents $x_1 \ldots x_n$.

To represent *new* (as opposed to *given*) information, we add a single feature to the language of Discourse Representation Structures (DRSs). This feature is only applied to basic conditions. The extension to the DRS-syntax is as follows: if C is a basic condition, then NEW(C) is also a basic condition. Considering alternative semantic interpretations for new information in DRSs is not the concern of this paper, so we interpret such kind of conditions as were they of their original form, for a semantic interpretation function $[\![.]\!]$, we have $[\![\text{NEW(C)}]\!]=[\![\text{C}]\!]$.

(Note that this has a number of (perhaps unpleasant) consequences. Only that information which contributes to basic-conditions can be marked as new. This excludes, for instance, determiners and connectives. Also note that the NEW tag has a temporary status, and needs to be updated after each new sentence in the text or utterance in a dialogue.)

We consider two ways in determining new information in an ongoing discourse or dialogue. The first possibility is declaring accommodated material introduced by presupposition triggers such as definite noun phrases as new. The second possibility is using informativity checks on parts of newly uttered expressions.

### 4.2.2 Accommodation

Van der Sandt (1992) gives an *anaphoric* account of presupposition. That is, in his view presuppositions behave very much like anaphoric pronouns—in fact the only difference is that presuppositions have more descriptive content. These simple ideas have two important consequences. First, there is no need to give an account of presupposition 'cancellation', for there simply is no such phenomenon; what other accounts regard as a 'cancellation' is simply a case of a presupposition being successfully resolved to an antecedent. Second, because they have descriptive content, presuppositions are sometimes able to 'repair' the context by creating a suitable antecedent; this process is known as *accommodation* ( Lewis (1979) ).

Van der Sandt expresses his theory in DRT and lets presupposition triggers contribute a new DRS to the evolving representation, and demands that this picture be incorporated into the overall representation. Two incorporation mechanisms are permitted. First, presuppositions can be *resolved*, just like ordinary pronouns in DRT. Second, presuppositions can be *accommodated*; that is, they can repair the context by creating their own antecedent.

We assume that presupposition triggers in the lexicon (such as the definite article, possessive constructions, and proper names) determine what is presupposed information, and what is asserted. To link Van der Sandt's algorithm with our notion of new information, we alter it in such a way that basic conditions within accommodated material contain the feature NEW. Resolved material, on the other hand, is added to the DRS without providing the newness feature. After all, resolved information is old, and accommodated material is new to the hearer.

### 4.2.3 Informativity

One of the maxims of conversation is to provide *new* information in each contribution to a dialogue. Semantically, this can be tested by entailment; the information expressed in a new utterance should not follow from the previous information. Assume that $\phi$ represents the text so far, $\pi$ represents background information, and $\psi$ a new utterance, then the new utterance is *not* informative if $\phi \wedge \pi \rightarrow \psi$ is a theorem. This can also be applied to DRSs, by using the translation function from DRSs to first-order logic ( Blackburn *et al.* (1999) ). Suppose that:

$$\mathrm{x}_1 \ldots \mathrm{x}_m \mathrm{C}_1 \ldots \mathrm{C}_m$$

represents a discourse or dialogue, and that the DRS

$$\mathrm{y}_1 \ldots \mathrm{y}_n \mathrm{D}_1 \ldots \mathrm{D}_n$$

represents a new utterance continuing the former text. By using the translation function $(.)^{fo}$ from DRSs to first-order logic ( Blackburn *et al.* (1999) ) we can check whether

$$((\mathrm{x}_1 \ldots \mathrm{x}_m \mathrm{C}_1 \ldots \mathrm{C}_m)^{fo} \rightarrow (\mathrm{y}_1 \ldots \mathrm{y}_n \mathrm{D}_1 \ldots \mathrm{D}_n)^{fo})$$

is a theorem to detect informativity. If it is no theorem, then the new utterance is not informative, and we could extend the old DRS by the new DRS in which all basic conditions D are replaced by NEW(D) to signal new information.

Not all information within a new utterance needs to be new. (However, sometimes all the information in an utterance can be new, for example at the beginning of a conversation.) If an utterance is new (informative), then parts of it will represent new information, and the remaining is given. To determine new information *within* a new utterance, one can use the same technique on *parts* of an utterance. Partitioning the DRS K into a partial DRS K$'$ is done by taking the universe of K, and a member of the power-set of K's set of conditions (excluding the empty set). As DRSs are defined recursively, the portioning algorithm needs to mirror this recursive structure in its processing steps. This resulting partial DRSs need to be tested on informativity by the method describe above.

## 4.3 System Descriptions

### 4.3.1 MIDAS

MIDAS (Multiple Inference-based dialogue analysis system) was developed to study the role of automated reasoning in human machine dialogue systems. The overall system structure is

based on the Dialogue Move Engine architecture as proposed in the Trindi (Task Oriented Instructional Dialogue, EC Project LE4-8314) and implemented by TrindiKit ( Larsson *et al.* (1999) )).

On the analysis side, MIDAS uses a left-corner parser, a lexicon consisting of ca. 4000 inflected forms, and a phrase structure rule grammar (for English). The system's utterances are partly pre-canned, but mostly generated directly from the semantic representation, using the same lexicon and a subset of the grammar rules as used for analysis.

The scenario covers the 'route planning service' task: MIDAS, having primary initiative throughout the dialogue, asks the user for a destination, departure, and traveling time, and whether the quickest or the shortest route should be presented.

The semantic framework underlying MIDAS is Discourse Representation Theory ( Kamp and Reyle (1993) ). Semantic analysis includes resolution of scope, anaphora, ellipsis and presupposition. The information state of the dialogue is modeled by Discourse Representation Structures, extended with machinery to represent dialogue acts, and utterance grounding, similar to that as proposed by Poesio and Traum (1998).

In general, we would like to equip natural language processing systems (like MIDAS) with a reasoning component to react intelligently on the user's input. This requires coping with the content of the user's contribution, and a deep semantic analysis to deal with inconsistent information, or with more or less information than was requested. In MIDAS, first-order reasoning is used to resolve ambiguities (scope, anaphora, presupposition, ellipsis), and to interpret answers of users to questions posed by the system.

### 4.3.2 MathWeb

Most state-of-the-art inference engines don't work on DRSs directly. By using a translation approach from DRT to first-order logic we are able to use a wide variety of off-the-shelf provers. Note that inference problems need not be theorems—they can be satisfiable as well, and we do not know beforehand. A typical example is the check for consistency: $\phi$ is inconsistent if $\neg\phi$ is a theorem, and consistent if $\phi$ is satisfiable. A handicap is the undecidability of first-order logic. By using different inference engines (with different strategies) at the same time in a distributed framework, we reduce this gap to a minimum.

Automated theorem proving has seen an enormous increase of performance of (especially first-order) inference engines. We argue to farm out the inference tasks to many different provers simultaneously, by combining MIDAS with the distributed MathWeb theorem proving environment ( Blackburn *et al.* (1999); Franke and Kohlhase (1999) ), because typically a high number of reasoning (of rather "simple") tasks are generated by MIDAS; and there are significant differences in speed and coverage that state-of-the-art provers offer.

Using MathWeb and the Internet as medium to distribute the inference engines across different machines provides an ideal testbed for studying the role of automated reasoning in computational semantics in general, and in particular for dialogue systems such as MIDAS.

The arsenal of inference engines MIDAS currently calls upon include the theorem provers Bliksem, SPASS, Otter, FDPLL, and the model generator MACE.

### 4.3.3 Festival

MIDAS uses Festival for speech synthesis. The Festival system was developed by Alan Black and Paul Taylor, at the Centre for Speech Technology Research, University of Edinburgh ( Taylor *et al.* (1998a) ). One of is features is that it supports text input in Sable format.

Sable is a system non-specific mark-up language to annotate prosodic features in texts, and offers a well-defined way of marking up text so that the synthesizer may render it appropriately.

## 4.4 The Concept-to-Speech Component in MIDAS

### 4.4.1 Generating Expressions from DRSs

The generation component in MIDAS works on the basis of DRS input and string output. It uses the same lexicon as in the utterance analysis component. There are some fixed expressions that are hardwired with specific dialogue moves (such as basic answers to yes-no questions, greetings, apologizes). The grammar for the generator focuses on questions. Most rules are duplicates of the analysis grammar.

The generator has three input parameters: a dialogue act, an utterance DRS, and a context DRS. The output parameter is a string. On the basis of the dialogue act the generator decides the basic utterance type (e.g., whether it will be a check-question, a yes-no-question or wh-question). Then it incrementally traverses trough the DRS, and deletes information from the utterance DRS while applying the grammar and lexical rules. Generation of a list of words succeeds when after applying the grammar rules the utterance DRS is empty.

The generator uses the context of the dialogue. Given the DRS of the total information state of a point in the dialogue, the utterance DRS is the current discourse unit under discussion, and the context DRS is the remaining DRS. The context DRS serves to generate context-sensitive expressions such as proper names or definite descriptions, which appear as free variables in the utterance DRSs, but are bound by discourse referents in the context DRS.

Words that are generated from basic conditions that have the feature NEW, receive a label saying so. For questions, wh-words also receive this label. This establishes the interface that carries over given and new information found on the semantic level (in the DRS) to the actual string.

### 4.4.2 The Generation-Prosody Interface

The ultimate goal for focus ground determination is to use this information in the synthesis of the system's output. Strings need to be annotated by prosodic features. We use the following Sable tags:

<RATE SPEED="-30%"> *repeated information* </RATE>

<RATE SPEED="-60%"><EMPH> *new information* </EMPH></RATE>

These tags mark repeated information to be pronounced at a slower rate, and new information to be pronounced at an even slower rate plus an emphatic accent.

### 4.4.3 Examples of Concept-to-Speech in MIDAS

Consider the following snapshot of a dialogue with MIDAS (M: MIDAS, U: User). In this dialogue, MIDAS is set to pessimistic mood, making it generate a check-questions after a contribution of the user.

(4.1) M: <SABLE><RATE SPEED="-60%"><EMPH> when </EMPH></RATE> do you go ? </SABLE>

(4.2) U: I would like to go to Birmingham.

(4.3) M: <SABLE> did you say <BREAK LEVEL="SMALL"/> <RATE SPEED="-30%"> you go to <RATE SPEED="-60%"><EMPH> birmingham </EMPH></RATE> </RATE> ? </SABLE>

(4.4) U: I like that city.

(4.5) M: <SABLE> did you say <BREAK LEVEL="SMALL"/> <RATE SPEED="-30%"> you like birmingham </RATE> ? </SABLE>

In (4.1), MIDAS poses a question to the user, and as the generator in MIDAS assigns wh-words as new information, 'when' gets an accent. The user replies that (s)he wants to go to Birmingham. Birmingham is a proper name, and because it is not mentioned before in the dialogue, accommodated to the main DRS, and consequently marked as new information. Accordingly, in (4.3), 'birmingham' is stressed in the check-question. The user responds with a definite noun phrase 'that city', which is resolved to the earlier introduced city Birmingham. In the check-question (4.5), therefore, 'birmingham' is not stressed.

## 4.5 Related and Further Work

A similar system with a concept-to-speech component is MIMIC ( Nakatani and Chu-Carroll (2000) ). Compared to MIDAS, it is more sophisticated as is deals with several kinds of accents. On the other hand, the representations they work with are AVMs which values are highly system-dependent, whereas MIDAS uses general structures, and transferring the MIDAS technology to other systems is straightforward.

Although in most cases new information correspond with stressed constituents, there might be exceptions triggered by syntactic structure of the utterance. Positioning and addressing this focus-projection problem requires further work. Another potential area of research is to use the emotional state of MIDAS in concept-to-speech generation.

## URLs

**Festival** : `www.cstr.ed.ac.uk/projects/festival/`

**MIDAS** : `www.coli.uni-sb.de/~bos/midas/`

**SABLE** : `www.cstr.ed.ac.uk/projects/sable.html`

# Bibliography


Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., and Traum, D. R. (1995). The TRAINS project: a case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, **7**, 7–48.

Bard, E. G., Sotillo, C., Anderson, A. H., and Taylor, M. M. (1995). The DCIEM MapTask corpus: Spontaneous dialogue under sleep deprivation and drug treatment. In *Proc. of the ESCA-NATO Tutorial and Workshop on Speech under Stress*, Lisbon.

Berger, A., Della Pietra, S., and Della Pietra, V. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, **22**(1), 39–72.

Predicting the intonation of discourse segments from examples in dialogue speech. in Sagisaka, Y., Campbell, N., Higuchi, N. (Eds.),*Computing Prosody*. Springer, Berlin.

Blackburn, P., Bos, J., Kohlhase, M., and de Nivelle, H. (1999). Inference and Computational Semantics. In H. Bunt and E. Thijsse, editors, *Third International Workshop on Computational Semantics (IWCS-3)*, pages 5–19. Computational Linguistics, Tilburg University.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Chapman and Hall.

Brubaker, R. (1972). Rate and pause characteristics of oral reading. *J. Psycholinguistic Research*, **1**, 141–147.

Bruce, G. (1982). Textural aspects of prosody in Swedish. *Phonetica*, **39**, 274–287.

Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., and Anderson, A. H. (1997). The reliability of a dialogue structure coding scheme. *Computational Linguistics*, **23**(1), 13–32.

Chu-Carroll, J. (1998). A statistical model for discourse act recognition in dialogue interactions. In J. Chu-Carroll and N. Green, editors, *Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 12–17. AAAI Press.

Cooper, R., Engdahl, E., and Larsson, S. (2000). Accomodating questions and the nature of qud. In *Proc. of GTALOG 2000 - FOURTH WORKSHOP ON THE SEMANTICS AND PRAGMATICS OF DIALOGUE*, Gteborg.

Franke, A. and Kohlhase, M. (1999). System description: Mathweb, an agent-based communication layer for distributed automated theorem proving. In *16th International Conference on Automated Deduction CADE-16*.

Grosz, B. and Sidner, C. (1986). Attention, intentions, and the structure of discourse. In *Computational Linguistics*, **12**(1), 175–204.

Grosz, B. and Hirschberg, J. (1992). Some intonational characteristics of discourse structure. In *Proceedings of ICSLP-92*, Banff, 1992.

Hieronymus, J. and Williams, B. (1991). A comparison of the prosody in read speech and directed monologue in British English. In *Proceedings of the ESCA Workshop on the phonetics and phonology of speaking styles*, Barcelona, Spain 1991.

Hiyakumo, L., Prevost, S., and Cassel, J. (1997). Semantic and Discourse Information for Text-to-Speech Intonation. In *Proceedings ACL Workshop on Concept-to-Speech Technology*.

Hockey, B. A., Rossen-Knill, D., Spejewski, B., Stone, M., and Isard, S. (1997). Can you predict responses to yes/no questions? yes, no, and stuff. In *Proceedings of Eurospeech-97*, pages 2267–2270.

Jelinek, F. and Mercer, R. L. (1980). Interpolated estimation of Markov source parameters from sparse data. In E. S. Gelesma and L. N. Kanal, editors, *Pattern Recognition in Practice*, pages 381–397. North-Holland.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic*. Kluwer, Dordrecht.

King, S. (1998). *Using Information above the Word Level for Automatic Speech Recognition*. Ph.D. thesis, University of Edinburgh.

Larsson, S., Bohlin, P., Bos, J., and Traum, D. (1999). TRINDIKIT 1.0 Manual. Technical Report Deliverable D2.2, Trindi.

Lehiste,I. (1980). The phonetic structure of paragraphs. in Nooteboom, S., Cohen, A. (Eds.),*Structure and Process in Speech Perception*. Springer, Berlin, pp. 195-206.

Levinson, S. (1983). *Pragmatics*. Cambridge University Press.

Lewin, I., Russell, M., Carter, D., Browning, S., Ponting, K., and Pulman, S. (1993). A speech-based route enquiry system built from general-purpose components. In *Proceedings of Eurospeech-93*, pp. 2047–2050.

Lewis, D. K. (1979). Scorekeeping in a language game. *Journal of Philosophical Logic*, **8**, 339–359.

Mikheev, A. (1998). Feature lattices for maximum entropy modeling. In *Proc. of ACL-COLING*, pages 845–848, Montreal, CA.

Nagata, M. and Morimoto, T. (1994). First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, **15**, 193–203.

Nakajima, S. and Allen, J. (1993). A study on prosody and discourse structure in cooperative dialogues. *Phonetica*, **50**, 197–210.

Nakatani, C. (1996). Discourse structural constraints on accent in spontaneous narrative. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, editors, *Progress in Speech Synthesis*, pages 139–156. Springer, New York.

Nakatani, C. and Chu-Carroll, J. (2000). Using Dialogue Representations for Concept-to-Speech Generation. In *Proceedings of ANLP/NAACL Workshop on Conversational Systems*, Seattle, 2000.

Poesio, M. and Mikheev, A. (1998). The predictive power of game structure in dialogue act recognition: Experimental results using maximum entropy estimation. In *Proceedings of ICSLP-98*, Sydney, 1998.

Poesio, M. and Traum, D. (1998). Towards an axiomatisation of dialogue acts. In J. Hulstijn and A. Nijholt, editors, *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues*, pages 207–222, Enschede, Universiteit Twente, Faculteit Informatica.

Power, R. J. D. (1979). The organization of purposeful dialogue. *Linguistics*, **17**, 107–152.

Provost, S. and Steedman, M. (1994). Specifying intonation from context for speech synthesis. *Speech Communication*, **15**(1-2), 139–153.

Rabiner, L. and Juang, B.-H. (1994). *Fundamentals of Speech Recognition*. Prentice Hall.

Reithinger, N. and Klesen, M. (1997). Dialogue act classification using language models. In *Proc. of Eurospeech-97*, pages 2235–2238, Rhodes.

Rosenfeld, R. and Clarkson, P. (1997). *CMU-Cambridge statistical language modeling toolkit, v. 2*. CMU, Pittsburgh. Available at `http://svr-www.eng.cam.ac.uk/ ~prc14/`.

Shriberg, E., Taylor, P., Bates, R., Stolcke, A., Ries, K., Jurafsky, D., Coccaro, N., Martin, R., Meteer, M., and Van Ess-Dykema, C. (1998). Can prosody aid the automatic classification of dialogue acts in conversational speech? *Language and Speech*,**41** (3–4).

Siegel, S. and Castellan, N. J. (1988). *Nonparametric statistics for the Behavioral Sciences*. McGraw-Hill, 2nd edition.

Sluijter, A. and Terken, J. (1993). Beyond sentence prosody: Paragraph intonation in dutch. *Phonetica*, **50**, 180–188.

Swerts, M. (1994). Prosodic features of discourse units. monologues. Doctoral Dissertation, Eindhoven University of Technology (unpublished).

Swerts, M. (1997). Prosodic features at discourse boundaries of different strength. *JASA*, **101**, pp. 514-521.

Swerts, M., Geluykens, R. (1994). Prosody as a marker of information flow in spoken discourse. *Language and Speech*, **37**, 21–43.

Swerts, M., Collier, R. Terken, J. (1994). Prosodic predictors of discourse finality in spontaneous monologues. *Speech Communication*, **15**(1-2), 79–90.

Taylor, P. (2000). Analysis and synthesis of intonation using the tilt model. *JASA*, **107** (3), pp. 1697-1714.

Taylor, P., King, S., Isard, S., and Wright, H. (1998). Intonation and dialogue context as constraints for speech recognition. *Language and Speech*, **41**(3-4), pp. 493-512.

Taylor, P., Black, A., and Caley, R. (1998a). The architecture of the festival speech synthesis system. In *Proc. of the 3rd ESCA Workshop on Speech Synthesis*, pages 305–310.

Thorsen, N. G. (1985). Intonation and text in standard danish. *JASA*, **77**, 1205–1216.

Van der Sandt, R. A. (1992). Presupposition Projection as Anaphora Resolution. *Journal of Semantics*, **9**, 333–377.

Wright, H. (1998). Automatic utterance type detection using suprasegmental features. In *Proc. ICSLP-98*.

Wright, H. (1999). *Modelling Prosodic and Dialogue Information for Automatic Speech Recognition*. Ph.D. thesis, CSTR, University of Edinburgh.