Modelling Prosodic and Dialogue Information for Automatic Speech Recognition

Helen Frances Wright

Thesis submitted for the degree of Doctor of Philosophy

University of Edinburgh

1999

Declaration

I have composed this thesis. The work in it is my own unless otherwise stated.

Helen F. Wright

Acknowledgement

I am most grateful to my supervisor Stephen Isard for all his time and patience. I would also like to thank Bob Ladd, my second supervisor, for his guidance. I received much support from my colleagues at the Centre of Speech Technology Research, particularly Paul Taylor and Simon King. In addition, I acknowledge the contribution of Massimo Poesio.

I would like to thank my parents and Stuart for their support over the years.

I hold an EPSRC PhD studentship 9630715.

Abstract

There are two main goals of the work presented in this thesis. The first is to provide a system for automatically classifying utterances into different types known as *moves*. In order to do this, one can take advantage of certain constraints found in natural dialogues. Moves of the same type have similar syntax and intonation features. In addition, moves follow each other with a degree of regularity. This study joins together these three aspects to perform automatic move detection. The second goal is to use this move classification in an automatic speech recognition system to constrain the recognition candidates. This system is successful in identifying utterance types and subsequently reduces the word error rate of the recogniser. It also provides an in depth study into how people express the discourse function of an utterance through intonation and provides a model of dialogue structure at a number of levels.

Contents

1 Introduction **1**

1.1 Thesis Structure 3

1.2 Distribution of Work 5

1.3 Published Work 6

1.4 Applications 6

1.5 Spoken Language Systems 6

1.5.1 TRAINS 7

1.5.2 Verbmobil 9

1.5.3 Question-acknowledgement Statement Classifier 11

2 Dialogue Annotation Schemes **13**

2.1 Discourse Plans 14

2.2 Adjacency pairs 15

2.3 Speech Act Theory 16

2.4 Conversational Games 17

2.4.1 Moves 17

2.4.2 Initiating Moves 18

2.4.3 Response Moves 21

2.4.4 Games 23
2.4.5 Transactions 23
2.5 DAMSL 24
2.6 Switchboard Data Recognition 25
2.6.1 Clarity 27
2.7 Verbmobil 27
2.8 Dialogue Annotation of Classroom corpus 27
2.9 Annotation for a Japanese Dialogue System 28
2.10 Topic Structure Identification 29
2.11 Assessing Discourse Annotation Schemes 30
2.11.1 Annotation Accuracy of Game Analysis Theory 31
2.11.2 Agreement Results for Other Annotation Schemes 33

3 Experimental Setup **35**

3.1 Introduction 35
3.2 The Data 36
3.2.1 The Map Task Scenario 36
3.3 Choice of Discourse Annotation Scheme 36
3.3.1 Data Annotation 38
3.4 System Architecture 39
3.4.1 Automatic Move Classification 40
3.4.2 Automatic Speech Recognition 45
3.4.3 The Baseline System 45
3.4.4 Measuring Success 46

3.4.5 Testing Scenarios 46

4 Dialogue and Language Models 47

4.1 Introduction 47

4.2 Language Modelling Techniques 48

4.2.1 N-grams 48

4.2.2 Backing Off 50

4.2.3 Interpolation Techniques 51

4.2.4 Perplexity and Entropy 52

4.3 Language Modelling Experiments using the Map Task Data . . . 53

4.3.1 Experimental Setup 53

4.3.2 Perplexity Results 55

4.4 Dialogue Modelling 56

4.4.1 Top-down Approaches 57

4.4.2 Bottom-up Approaches 58

4.4.3 N-grams for Dialogue Modelling: Previous Work 60

4.5 Dialogue Modelling Experiments for the Map Task 61

4.5.1 Simple N-grams 61

4.5.2 Including Speaker Information 63

4.5.3 Results 64

4.6 Move Recognition Results 65

4.6.1 Conclusion 67

5 Automatically Extracting Intonation Features 68

5.1 Introduction 68
5.2 Intonation Analysis 69
5.2.1 IPO 69
5.2.2 British School of Intonation Description 70
5.2.3 Autosegmental-metrical Theory of Intonation 71
5.2.4 ToBI 72
5.2.5 Verbmobil 73
5.2.6 Fujisaki 73
5.2.7 The Tilt Model 74
5.3 Automatic Analysis of the Intonation Contour 77
5.3.1 Automatic Event Detection 78
5.3.2 Aligning the Rise and Fall Parts of Events 79
5.3.3 Automatic Tilt Analysis 79
5.4 Other features 81
5.4.1 F0 features 81
5.4.2 Energy features 83
5.4.3 Duration Features 83
5.5 Summary 84

6 Intonation Models **85**

6.1 Introduction 85
6.2 Mapping Intonation to Discourse Function 86
6.2.1 The Top-down Approach 86
6.2.2 The Bottom-up Approach 86

6.3 Statistical Models of Intonation 88
6.4 Classification and Regression Trees 89
6.4.1 Previous use of CART Trees in Synthesis and Recognition of Intonation 89
6.4.2 Training CART Trees 91
6.4.3 Experiments Using the DCIEM Data 92
6.4.4 Tree Interpretation 93
6.5 Neural Nets 97
6.5.1 Appropriate Problems for ANNs 97
6.5.2 The Perceptron 98
6.5.3 Training the Models 99
6.6 Hidden Markov Models 102
6.6.1 Finite State Networks as Intonation Model 102
6.6.2 Training the Models 104
6.6.3 Discussion 106
6.6.4 Resynthesising the Intonation Contour using HMMs 107
6.7 Move Detection Results using Intonation 113
6.8 Summary 117

7 System Performance **119**

7.1 Comparison with Similar Speech Recognition Systems 120
7.2 Summary of DCIEM Move Recognition Results 121
7.3 Word Recognition Results 124
7.4 WER using Different Intonation Models 127

7.5 Discussion 128
7.6 Clustering and Splitting the 12 Moves 129
7.6.1 Context Dependent Moves 129
7.6.2 Results of Merging and Splitting 130

8 Using Game Information to Improve Move Recognition 133

8.1 Introduction 133
8.2 Automatically Identifying Game information 134
8.3 Chapter Structure 137
8.4 The Data 138
8.4.1 Data Analysis 139
8.5 Dialogue Models 142
8.5.1 Modelling Game and Move Information Simultaneously . . 144
8.5.2 Comparing ME with N-grams for Modelling Game Structure 147
8.6 Intonation Models 150
8.7 Language Models 152
8.7.1 Smoothing Move_position Language Models 153
8.8 Modifying the Move_position Utterance Type Set 155
8.8.1 Language Models for Move_position Set 2 157
8.8.2 Dialogue Models for Move_position Set 2 159
8.8.3 Intonation Models for Move_position Set 2 159
8.8.4 Move_position Set 2 Recognition Results 162
8.8.5 Word Recognition using Move_position set 2 162
8.8.6 Original Move recognition Results 164

8.8.7 Word Error Rate 166
8.8.8 Declarative and Interrogative Recognition 166
8.9 Summary 170

9 Automatically Predicting the Syllabic Peak Position 171

9.1 Peak Position 171
9.2 Literature Review 174
9.3 Predicting Peak Position 176
9.3.1 The Data 176
9.3.2 Training the Tree 177
9.4 Results 178
9.5 Tree Interpretation 181
9.6 Conclusion 182

10 Conclusion and Future Directions 184

10.1 The Goals 184
10.1.1 Utterance Type Recognition 184
10.1.2 Word Recognition 185
10.2 Areas of further development 185
10.2.1 Discourse Annotation 185
10.2.2 Utterance Type Recognition 186
10.2.3 Intonation Models 187
10.2.4 Dialogue Models 187
10.2.5 Language Models 188

10.2.6 Discourse Markers 189

10.3 Conclusion 190

References **191**

List of Figures

1.1 Thesis Structure 4

2.1 Move categorisation. Source: Carletta *et al.* (1997) 19

3.1 Example of two slightly different maps given to the Map Task participants 37

3.2 System Architecture 41

4.1 Move bigram finite state network; not all arcs are given 58

4.2 Probabilistic move bigram finite state network; not all arcs are given 59

4.3 Markov model of dialogue with transitional probabilities. Source: Woszczyna and Waibel (1994) 60

4.4 4-gram for move sequence modelling 62

4.5 Dialogue Model III 64

5.1 Division of the intonation contour by the British School. Source: Ladd (1990) 70

5.2 FSN for all possible event combinations. Source: Pierrehumbert (1980) 72

5.3 FSN for all possible event combinations. Source: Ladd (1996) . . 72

5.4 Values for tilt for various shaped intonation events 75

5.5 Rise fall accent showing the tilt parameters, excluding tilt 76

5.6 Intonation contour with labelled accents and boundaries corresponding to the circled pitch excursions 78

5.7 a) shows F0 smoothing; b) rise/fall fitting using the RFC model; c) amplitude and duration calculations used in the tilt analysis. Source: Taylor *et al.* (1998). 80

5.8 Intonation contour with least square regression lines capturing the line of declination and the boundary tone 83

6.1 Part of the binary decision tree for classifying moves 95

6.2 Example contour of a yes-no question move that has a falling boundary ("fb") and a high accent on the nuclear accent ("a") on "monastery". ("m" is a minor accent, see section 5.2.7) 96

6.3 A single perceptron 99

6.4 Intonation structure represented by finite state networks 103

6.5 A three state, left-to-right HMM 104

6.6 Highest weighted Gaussian mixture component for the tilt values of the third state of the *query-yn* HMM 108

6.7 Four F0 contours with automatic intonation labelling and words. From top down: original F0; resynthesised F0 from original tilt parameters; two synthesised contours from a yes/no query HMM . 111

8.1 Chapter overview 136

8.2 Model F for move_position modelling 147

8.3 Model VII giving 4.6 test set perplexity 149

8.4 Model VIII giving 4.6 test set perplexity 150

8.5 Model IV used in ME and 4-gram experiments 151

8.6 Percentage of interrogative and declarative type utterances correctly recognised 167

9.1 Contour schema showing ideal peak position measurement 172

9.2 Intonation contour with labelled accents and boundary corresponding to the circled pitch excursions. Each accent is linked to a stressed syllable. Source: Taylor (2000). 173

List of Tables

2.1 Data extract including move and game type 18

2.2 Examples of the five most frequent dialogue acts after clustering, with the percentage of the data they account for 26

3.1 Statistics for training and testing sets of the DCIEM corpus . . . 38

3.2 Hand-labelled training and testing sets of the DCIEM corpus . . . 38

3.3 Frequency of move types for set B training set, follower's moves and giver's moves 39

4.1 Move-specific LM training set sizes 54

4.2 The interpolation weights of the move-specific language models . . 55

4.3 Language model perplexities 56

4.4 Perplexities of simple N-gram dialogue models. Source: King (1999) 62

4.5 Notation of N-gram candidate predictors 64

4.6 Perplexity results for the different dialogue models 65

4.7 Move detection results using various information sources in the overhearer scenario 65

5.1 F0 feature list 82

5.2 Energy feature list 84

5.3 Duration feature list 84

6.1 Discriminatory features and type usage in move classification trained using equal number of moves 94

6.2 Top 10 features and feature type usage 94

6.3 Percentage of moves correctly recognised using the intonation models (IM) in conjunction with various dialogue models and the language models and recogniser (REC) 114

6.4 Percentage of initiating and non-initiating moves correctly recognised 114

6.5 Move recognition results for CART tree trained on original data. 44.7% of the moves are correctly classified. 115

6.6 Move recognition results for CART tree trained on data of equal move types. 31.2% of the moves are correctly classified. 116

6.7 Percentage of declaratives and interrogative type moves correctly classified by the different types of CART tree 117

7.1 Move detection results using various information sources in the overhearer scenario 121

7.2 Confusion matrix for move type classification: 64% move recognition accuracy 122

7.3 Percentage of declaratives and inter correctly classified by CART tree trained on equal data 123

7.4 Move detection and WER results using various information sources in the overhearer scenario 124

7.5 System performance compared with baseline for initiating and non-initiating moves 126

7.6 System performance comparing different intonation modelling techniques 127

7.7 Modified move types 130

7.8 Move detection and WER results using various information sources in the overhearer scenario 131

8.1 Data extract including game, position and move type 139

8.2 DCIEM Map Task data statistics for training set B 140

8.3 Glasgow Map Task data statistics 140

8.4 Distribution and average length of games in training set B 141

8.5 Frequencies of the most common initiating and the two most common non-initiating moves in various types of games in training set B 142

8.6 Dialogue model perplexities for DCIEM corpus 143

8.7 Dialogue model perplexities for Glasgow Corpus 143

8.8 Move frequencies with respect to game position 146

8.9 Perplexity results for the different dialogue models predicting move_position categories 146

8.10 Notation of N-gram predictors 148

8.11 Perplexity results for the different dialogue models for predicting original move types 148

8.12 Percentage of original moves correct using dialogue model (DM), intonation (I) and recogniser output (REC) in the transcription scenario 150

8.13 Utterance type detection using bigrams and intonation models in the transcription scenario 151

8.14 Perplexity results for the test set using the various sets of language models trained on set B 153

8.15 LM perplexity results for whole test set using different types of language models for the different utterance type sets 154

8.16 Smoothing weights towards the move_position and move specific language models 156

8.17 Perplexity results for the test set trained on set B, smoothed with set B general 157

8.18 Smoothing weights towards set 2 and move specific language models 158

8.19 Perplexity results for the different dialogue model 159

8.20 Recognition results for move and set 2 using Model E and intonation models in the overhearer scenario 159

8.21 Confusion matrix for move type classification: intonation 30% recognition accuracy 161

8.22 Percentage of declaratives and interrogative type moves correctly classified by CART tree trained on all data 162

8.23 Move detection results using various information sources in the overhearer scenario 163

8.24 Move detection accuracy using various information sources in the overhearer scenario 164

8.25 Confusion matrix for move type classification: 66% move recognition accuracy 166

8.26 System performance compared with baseline 168

8.27 Move detection and WER results using various information sources in the overhearer scenario 169

9.1 List of features used to train the regression tree to predict peak position 177

9.2 Correlation between real peak position and output of the regression tree using the BU corpus for training and testing 179

9.3 Correlation between real peak position and output of the regression tree using the KED corpus for training and testing 180

9.4 Correlation between real peak position and output of the regression tree using the KED corpus for training and BU corpus for testing 180

9.5 Correlation between real peak position and output of the regression tree using the BU corpus for training and KED corpus for testing 180

9.6 Correlation results for just accents using accents and boundaries to train the regression tree 181

9.7 Correlation results for accents and boundaries using accents and boundaries to train the regression tree 181

9.8 Discriminatory features and type usage in peak position prediction 182

Chapter 1

Introduction

There are two main goals of the work reported in this thesis. The first is to be able to model and automatically detect discourse structure. The second is to integrate this into an automatic speech recognition system to improve word recognition.

The first issue addressed is the automatic classification of utterances into different types, for example statements, question and replies. The term *utterance type* is used here and encodes the role of an utterance in the dialogue also known as its *dialogue act*. Automatic utterance type detection is performed by taking advantages of regularities in the following three areas:

- Utterances of the same type have similar syntactic patterns. For example in the Map Task, a yes-no question frequently starts with "Do you have...?".

- Utterances follow each other with a degree of regularity. For example, a query followed by a reply followed by an acknowledgement is more likely than three replies in a row.

- Utterances have distinguishing intonation patterns. For example, a sentence with a declarative syntax can be realised as a question or a statement depending on whether the utterance final intonation contour is rising or falling.

Previous studies have refrained from using intonation to distinguish utterance types as there is not a one-to-one mapping between intonation contour and discourse function. For example, a yes-no question frequently has a rising boundary but may have a falling boundary. I propose a solution to this problem by training *stochastic* models that can cope with the variation of intonation contours associated with one utterance type. Before this is possible, one has to extract the potential intonation features automatically from the data.

Twelve language models (LM) are trained for each of the different utterance types. These models are used to give the likelihood that a sequence of recognised words is an utterance of a certain type. Regularities in the sequences of utterance types are captured by a statistical dialogue model (DM). These dialogue models use discourse information such as the previous utterance type and speaker identities to predict the current utterance type.

The final section of this thesis shows that using information about the current discourse goal and where the participants are in achieving this goal can increase the predictability of the three statistical models described above. For example, utterances that introduce a new topic may be more emphatic than say an acknowledgement at the end of the dialogue. It will be shown that using this higher level discourse information significantly increases the utterance type recognition accuracy of the system.

If the system knows the type of an utterance, it has a greater chance of guessing what the words are. For example, questions frequently contain words such as "which, where, how". Integrating an automatic utterance type detector into a speech recogniser produces a reduction in the word error rate (WER).

1.1 Thesis Structure

The second half of this chapter gives a review of current spoken language systems that use dialogue information to some extent. One of the main problems the developers of these systems face is deciding on a dialogue analysis scheme that is expressive enough to cover all the observed phenomena yet succinct enough to avoid overgeneration. Current discourse analysis methods are discussed in chapter 2.

Chapter 3 gives a breakdown of the utterance type recognition system and how this is incorporated into the automatic speech recognition system. The choice of data and the discourse analysis theory used in the experiments are also discussed.

The rest of the thesis is mostly concerned with training statistical models for the three aspects of discourse described above: dialogue structure, syntactic or language modelling and intonation modelling. The structure of the main part of the thesis is illustrated in figure 1.1. Dialogue and language modules are dealt with together in chapter 4, as similar language modelling techniques are used to train both models.

The main part of the original work is reported in chapters 5 and 6. Chapter 5 looks at possible intonation features that can be used to train the statistical intonation models described in chapter 6. Methods of automatically extracting these features are discussed, specifically the tilt theory described in Taylor (2000). There are some issues concerning the tilt features which will be addressed in chapter 9. Specifically, a method is given for statistically modelling the alignment of the peak of an accent.

In chapter 6, three methods of statistical modelling of intonation are examined and compared: classification and regression trees, artificial neural nets and hidden Markov models.

Utterance type recognition results for the whole system are given in chapter

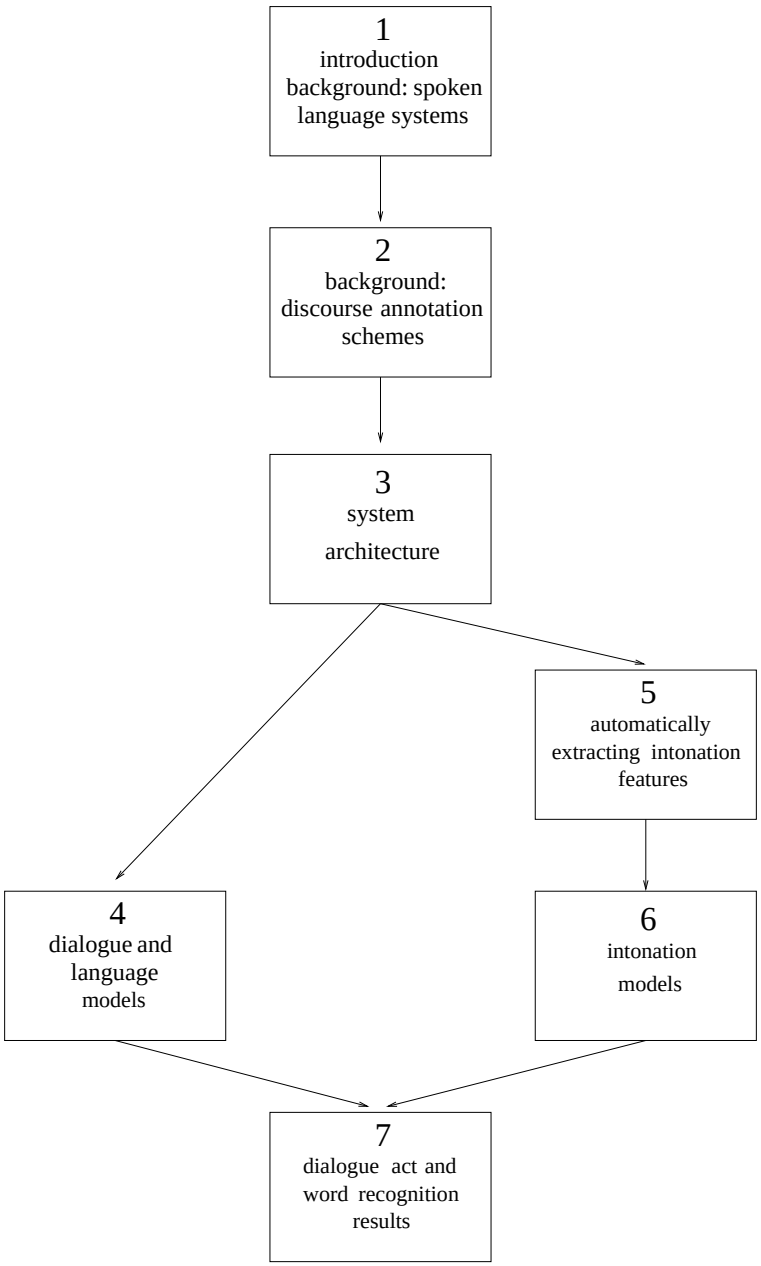


Figure 1.1: Thesis Structure

7. The type of an utterance, as classified by the system, is used in the automatic speech recognition system. The word recognition results show a significant improvement over the baseline using the system described in this thesis. Chapter 8 gives a method of improving the utterance type recognition results further by using higher level discourse information such as the current discourse goal and the stage one has reached in achieving this goal.

Finally, chapter 10 gives an summary of the work presented in this thesis and suggests areas of future investigation.

1.2 Distribution of Work

The system described in this thesis is the result of collaboration between a number of people. The automatic accent detector described in chapter 5 was developed by Paul Taylor (Taylor, 2000; Taylor, 1998). The work on dialogue and language modelling described in chapter 4 is that of Simon King (King, 1998). Stephen Isard also contributed to the work by personal communication. All other work, unless otherwise specified, is that of the author.

Most of this work is a result of an EPSRC[1] funded project (called ID4S) from October 1993 to March 1997. This project looked at using intonation and dialogue context in speech recognition. Extensions of this work by the author include examining various stochastic models for intonation modelling (chapters 5 and 6) and looking at issues arising from the automatic extraction of intonation features (chapter 9). Other original work includes using game information for utterance type detection (chapter 8), which is inspired by the work of and personal communication with Massimo Poesio (Poesio & Mikheev, 1998).

The main word recognition was performed using HTK (Young *et al.*, 1996). The software for the classification and regression tree was written by Alan Black

[1]Engineering and Physical Science Research Council

(Taylor *et al.*, 1998a). Other software used includes the Stuttgart neural net simulator (SNNS, 1997) and the CMU toolkit (Rosenfeld & Clarkson, 1997).

1.3 Published Work

The method of statistically modelling intonation is outlined in Wright and Taylor (1997) and Wright (1998) and described in full detail in chapters 5 and 6. General publications involving the ID4S project include Taylor *et al.* (1997) and Taylor *et al.* (1998b). Modelling higher level discourse information is reported in Wright *et al.* (1999) and discussed in full detail in chapter 8.

1.4 Applications

Utterance type detection is useful in human-computer interaction systems, for example the system needs to know if it is being asked a question or given a reply. The system also needs to know when a goal has been achieved so that it can update its knowledge base and move on to the next goal. A review of current spoken language systems that incorporate dialogue and prosodic information is given in the following section.

1.5 Spoken Language Systems

The goal of spoken language systems is to hold a task-orientated natural language dialogue with a human in a limited domain. Automatic dialogue analysis has three functions in these systems. Firstly, it can be used to improve word recognition. Secondly, if the system knows the type of utterance, recognising 100% of the words correctly is not always essential for understanding. For example, if one knows a positive reply has been uttered, the system does not have to bother distinguishing between the different types of replies such as "yeah, yep, right". Finally, a dialogue

manager component can use discourse information to aid semantic and pragmatic analysis.

Due to the complexity of the task the systems are very large with many different modules including a speech recogniser, natural language parser, proposition extractor, language generator and optionally a speech synthesiser. Modules which use discourse information will be discussed here, namely the dialogue manager and speech recognition components.

1.5.1 TRAINS

TRAINS (Allen *et al.*, 1996) is a system that conducts a conversation either through speech or typing and attempts to plan the desired train route between American cities. The planner is programmed so that it does not find a direct route thus inducing more elaborate dialogue.

The main components of interest here are the speech recogniser, a chart parser which determines the dialogue act[2] and a discourse manager. Dialogue act classification is an integral part of this system. The output of the parser and input to the discourse manager is a sequence of dialogue acts rather than a syntactic analysis. This forces an emphasis on semantic and pragmatic interpretation.

The vocabulary of this system is 1000 words. The speech recogniser used is a non-domain specific off-the-shelf recogniser, Sphinx-II system from CMU (Huang *et al.*, 1993). This system has a baseline word error rate of 20%.

The dialogue act annotation scheme adopted in this system is DAMSL. This scheme is described in detail in section 2.5.

- *Dialogue Act Classification*

[2]The term "dialogue act" is used here instead of "utterance type" to be consistent with the relevant literature

The automatic classification of dialogue acts (Hinkelman, 1990; Traum & Hinkelman, 1992) is performed in two processes. Firstly, the output of the recogniser is passed through a robust parser for syntactic and semantic analysis. A list of candidate dialogue act interpretations and their propositional content is derived from a number of rules based on syntactic and semantic properties. For example, "Can you do X?" can map to either a *request* or a *yes/no question* if taken literally. "Why not do X?" is mapped to a *suggestion* act. A second process examines the context and prunes the number of possible interpretations. These filters are based on checking the preconditions of the dialogue acts with the current knowledge state.

The use of dialogue act identification in a dialogue system is illustrated below (taken from Allen *et al.* (1996)). The badly recognised utterance "Okay now I take the last train in go from Albany to is" is divided into 3 dialogue acts

- a *confirm/acknowledge* ("okay")
- a *tell* with content to take the last train ("now I take the last train")
- a *request* to go from Albany ("go from Albany")

The correct transcription is "Okay now let's take the last train and go from Albany to Milwaukee". The dialogue act analyser has wrongly identified the second act but the dialogue manager has enough information to establish a confirmation and that a request has been made to move a train from Albany. The system continues the dialogue by starting a clarification sub-game.

- *Dialogue Manager*

The dialogue manager breaks up into a number of functions including dialogue act interpretation, planning, problem solving and domain reasoning. The sequence of dialogue acts is interpreted by the dialogue manager as illustrated in the above

example. This module keeps track of the discourse state as a stack. Each element of the stack captures the focus topic, history list and goal of the discourse segment and its status, i.e. whether it has been achieved or not.

- ***Other Train Enquiry Systems***

Baggia *et al.* (1997) and Eckert *et al.* (1996) report work using discourse structure in similar train enquiry dialogue systems for Italian and German respectively. Both systems decide on the user's utterance type depending on the system's own utterance. Most of these utterances are standard requests, such as a request for a specific time. This method of utterance type detection assumes a degree of cooperation of the speaker which, if, violated results in an incorrect interpretation. The predicted utterance type is used to improve word recognition. If one knows the type of an utterance, one can have a better guess at the words. For example, a reply to a request for a time is likely to include numbers. Details of word recognition results are given in section 7.1.

1.5.2 Verbmobil

The Verbmobil project (Wahlster, 1993) is a speech-to-speech translation system with the application of scheduling. There are two participants who both have a working knowledge of English. Automatic translation is available on request. The system has to keep track of the dialogue in order to catch certain contextual aspects such as reference. The system consists of 20 modules, a number of which use dialogue acts and prosody. The automatic prosodic annotation module is described chapter 5. A review of the Verbmobil dialogue model component is given in chapter 4.

The dialogue act tag set used in this system comprises of 43 acts which describe intentional content and the proposition of an utterance to a certain extent. This set is hierarchical and can be collapsed to a set of 18 primary intentions such

as *suggest, initiation* and *acceptance*. Further details of the tag set are given in chapter 2.

Automatic dialogue act identification, described in Warnke *et al.* (1997), is performed using two different methods. In the first, dialogue act segmentation is performed prior to classification. In the second, it is performed simultaneously. Various experiments use either hand-labelled word sequences or the recogniser output.

A multi-layer perceptron (MLP) is trained for dialogue act segmentation using 117 prosodic features for each word-final syllable. These prosodic features include duration, pause, F0-contour and energy. The MLP looks at a frame of six word-final syllables and outputs either $\tilde{D}$ or D for the dialogue act boundary prediction. This was improved upon by combining the MLP with a language model trained on data with inserted dialogue act boundary tokens between the words. Classification is subsequently performed by running the word sequence through language models that capture the syntactic properties of each of the dialogue act types (dialogue act specific language models). The dialogue act likelihoods from these language models are combined with probabilities from a dialogue model that looks at the previous dialogue act type within a speaker's turn.

The second method performs dialogue act classification and segmentation simultaneously by combining and weighting likelihoods from the various models: prosody, dialogue model and dialogue act specific language models. Therefore, at each word boundary a likelihood is given for a dialogue boundary and the word chain between the potential dialogue act boundaries is used to predict the utterance type.

Both methods produce similar results of approximately 53% accuracy, using the recognised word sequence. Running the system with the dialogue model only results in a slight decrease in error. This is not surprising as the dialogue model is restricted to using utterances in the same turn. Chapter 4 looks at more so-

phisticated dialogue models that make use of other information sources such as speaker identity.

1.5.3 Question-acknowledgement Statement Classifier

A simple dialogue act detection system is described in Terry *et al.* (1994) for use in a street-map directions dialogue system. The goal of this module is to distinguish between acknowledgements and questions using prosody. Distinguishing dialogue act types without prosody could lead to the misclassification of utterances. For example (taken from Terry *et al.* (1994)), the response to the system's instruction "turn right at the main street" could be any of the following:

- do I turn right at the main street?
- so RIGHT? at main
- right at MAIN?
- Okay, right at main

Apart from the first utterance which has interrogative syntax, prosody is needed to determine the utterance type of the replies. They use a set of 10 rules to map the intonation contour onto the utterance type. Firstly, they extract a set of prosodic features from the smoothed F0 contour. These features include duration, average pitch value and variance, pitch slope, goodness of fit, and maximum and minimum pitch values. These are then used to categorise parts of the contour into three shapes: convex, concave and straight. These shapes are classed as either rise, fall or level. Section 5.2.7 describes a system that captures the shape of a contour as a single continuous variable *tilt*, thus rendering these discrete labels redundant.

The set of 10 rules they present are based on two generalisations:

1. falling pitch indicates a statement

2. utterance final rising pitch indicates a query

For example, Terry *et al.* expand rule 2, saying that if there is a rising pitch that levels off at the end then the utterance is still classified as a query. Another example rule classes intonation contours that have a sharp rise at the beginning of the sentence and level off as a query or denoting uncertainty. Word spotting was also used for acknowledgements as they may have a rising boundary due to an element of doubt, for example "uh-huh", "go-on" and "okay". This method distinguishes queries from acknowledgements 89% of the time[3]. They have no way of recognising a query that has a falling intonation contour even if it has interrogative syntax. A novel method is presented in this thesis that uses statistical techniques that can map various intonation contours onto one utterance type.

[3]They do not give a figure for chance.

Chapter 2

Dialogue Annotation Schemes

The problem of which discourse analysis method to adopt is a major design decision in the construction of any dialogue system or automatic speech recognition system. One needs an annotation scheme that is large enough to be expressive yet succinct enough to be reliably coded. For the purpose of our experiments described in chapter 1, each utterance type must have a distinctive syntactic and prosodic form and yet be linguistically and pragmatically meaningful.

Two approaches are adopted throughout the literature: a *shallow* discourse structure and a *deep* discourse structure. Shallow approaches examine utterance form and discourse function in a small window of dialogue, such as speech act based utterance types (Austin, 1962; Nagata & Morimoto, 1993; Reithinger *et al.*, 1996). Other shallow analyses look at sociolinguistic facts such as appropriate replies captured by Schegloff and Sacks' adjacency pairs (Schegloff & Sacks, 1973). One deep structure approach looks at topic or focus structure identification (Nakajima & Allen, 1993). The theory adopted for this work is the *Conversational Game Analysis*, first proposed by Power (1979) and adapted for Map Task by Carletta *et al.* (1997). This theory takes a deep structure approach captured in plan based systems that classify utterances in terms of high-level discourse goals.

2.1 Discourse Plans

Plan based dialogue analysis schemes were developed (Power, 1979; Houghton, 1986) for use in computer-computer dialogue systems for generation and analysis of task oriented dialogues. Their schemes are hierarchical in structure and examine general goals of discourse and how these goals are achieved using different types of plans. Schemes developed by Power (1979) and an adaptation of this work by Houghton (1986) will be discussed here as their ideas form the basis of the Conversational Game Analysis adopted in this study.

Power's motivation for developing a plan based scheme was to gain insight into how utterances achieve goals. His practical application was to program two processes on one computer to communicate in such a way that they could achieve simple practical goals in a limited world, such as opening a door and entering. For the programs to cooperate, they have to exchange information, formulate plans, compare belief states and assess results, just as humans do.

The actual form of the conversation is determined by a number of procedures called *games*. In each of these games the agents take different roles. The games can be nested, which would occur if a subgoal needs to be achieved before the main one. The types of games defined by the program are listed below:

1. **GAME ASK** to obtain information
2. **GAME TELL** to give information
3. **GAME RULE** to discuss rules
4. **GAME GOAL** to ask for help with a goal
5. **GAME PLAN** to agree on a plan
6. **GAME ASSESS** to assess the result of an action

7. **GAME GAME** to get one of the other games started

Houghton (1986; 1987) defines a discourse unit of *interaction frames* which are similar to Power's games and are also designed to be used by agents in a virtual world. The types of interaction frames are given below:

1. **MAKE_KNOWN** impart information
2. **FIND_OUT** obtain information
3. **GET_DONE** get a favour done
4. **GET_ATTENTION** call someone

2.2 Adjacency pairs

Schegloff and Sacks' (1973) conversational structure is founded mostly on *adjacency pairs*. An adjacency pair consists of two utterances each of which is made by different speakers. For example: greeting-greeting; question-answer; offer:acceptance/refusal. Most types of utterances are specific to one part of the adjacency pair. The second part has two functions either to give a response or to acknowledge understanding. Despite the term *adjacency pair* the related utterances may have intervening utterances. For example a) and d) constitute an adjacency pair of suggest-agree with a clarification sub-pair in between.

- A: let's go to the cinema
- B: what's on?
- A: a cartoon
- B: all right

Power (1979) suggests that there is strong evidence that the adjacency pair unit is more important than the utterance as a dialogue unit. The whole point of uttering a question or an answer is to inform the listener, this cannot be achieved to the full extent without the adjacency pair context.

2.3 Speech Act Theory

Speech act theory (Searle, 1975; Austin, 1962), analyses the speaker's intention, meaning and beliefs. The basic premise behind this theory is that utterances can perform actions; such utterances are known as *performatives*. Actions such as performing a marriage can only be performed in the correct *felicity conditions*, for example if one is a priest or a judge. Austin revised this classification of utterances into performatives and constatives (statements) because an action can be performed without using a performative verb and conversely an intended action may not necessarily be achieved. The idea of performing was replaced by the notion of *illocutionary force*, which is the speaker's intention in uttering an act. The other aspects of an utterance are *locution* and *perlocution* described below:

- *locution*: uttering the words (phonemes or syllables)
- *illocution*: the intention behind the words (e.g. promising, suggesting, advising, etc.)
- *perlocution*: the effect of the illocution on the hearer (e.g. persuading, urging, etc.)

An utterance can be described on all the three levels given above. For example, take the utterance "You can't do that!". Locution is the utterance of the phonemes or words; the illocutionary act is either forbidding or protesting; and the perlocutionary effect could be to stop the addressee's action, annoy them or even incite them.

2.4 Conversational Games

This aspect of the effect of utterances is emphasised in the plan based schemes described above, where the participants only communicate when they need to perform some goal. It is the grouping of utterances with similar aims that motivated the development of the *Conversational Game Analysis* theory. Carletta *et al.* (1997) modified this theory of games for the type of dialogue created by the Map Task which is used in the experiments presented in this thesis. The Map Task is conducted by two people, one participant (the *giver*) has the role of guiding the *follower* around a map (more details are given in section 3.2).

The annotation scheme consists of three levels. The largest division of the dialogue is a *transaction* which involves the completion of a major part of the participant's plan. In the Map Task, this generally corresponds to the completion of some part of the route on the map. Transactions are subdivided into *games*. Dialogue games consist of an initiating utterance and encompasses all subsequent utterances that contribute to the aim of the game. Games consist of a sequence of *move* utterance types divided into initiations and responses. Each of these levels is discussed in turn in the sections below.

2.4.1 Moves

To a certain extent, move typing involves a shallow analysis by describing the utterance in terms of its form (e.g. *query-yn/query-w*) or its function in discourse (e.g. *ready, check*). The advantage of the move types described in Carletta *et al.* (Carletta *et al.*, 1997) is that they reflect the game structure. This is exemplified in figure 2.1 taken from Carletta *et al.* (1997), which shows the distinctions used to classify the move types. This figure shows that the move types are divided into three basic classes: *initiation* moves that start games, *response* moves and a preparation move *ready*. A breakdown of each of these move types is given in

Line	Speaker	Utterance	Move	Game	Game#
1	*Giver*	okay	*ready*	query-yn	1
2	*Giver*	at the starting point do you have a sandy shore?	*query-yn*	query-yn	1
3	*Follower*	yes I do	*reply-y*	query-yn	1
4	*Giver*	and directly below that do you have a well?	*query-yn*	query-yn	2
5	*Follower*	well?	*check*	check	embedded
6	*Follower*	no well	*reply-n*	query-yn	2
7	*Giver*	anything directly below?	*query-yn*	query-yn	3
8	*Follower*	hills way below	*reply-w*	query-yn	3
9	*Giver*	the well	*acknowledge*	query-yn	3
10	*Giver*	okay	*ready*	instruct	4
11	*Giver*	I'd say about three fingers south go down	*instruct*	instruct	4
12	*Follower*	of sandy shore, there's a well?	*check*	check	embedded
13	*Giver*	yeah	*reply-y*	check	embedded
14	*Follower*	okay	*acknowledge*	instruct	4

Table 2.1: Data extract including move and game type

Carletta *et al.* (1997) and is repeated here.

2.4.2 Initiating Moves

- ***The** Instruct **Move***

This type of utterance requests or demands an action. *Instruct* moves can be in the form of direct questions, imperatives or indirect suggestions. Due to the nature of the Map Task most instructions are given by the giver and mostly involve him guiding the follower around landmarks along the given route on the map. An example of an *instruct* is given in line 11 in table 2.1, where the giver guides the follower three fingers south to a landmark.

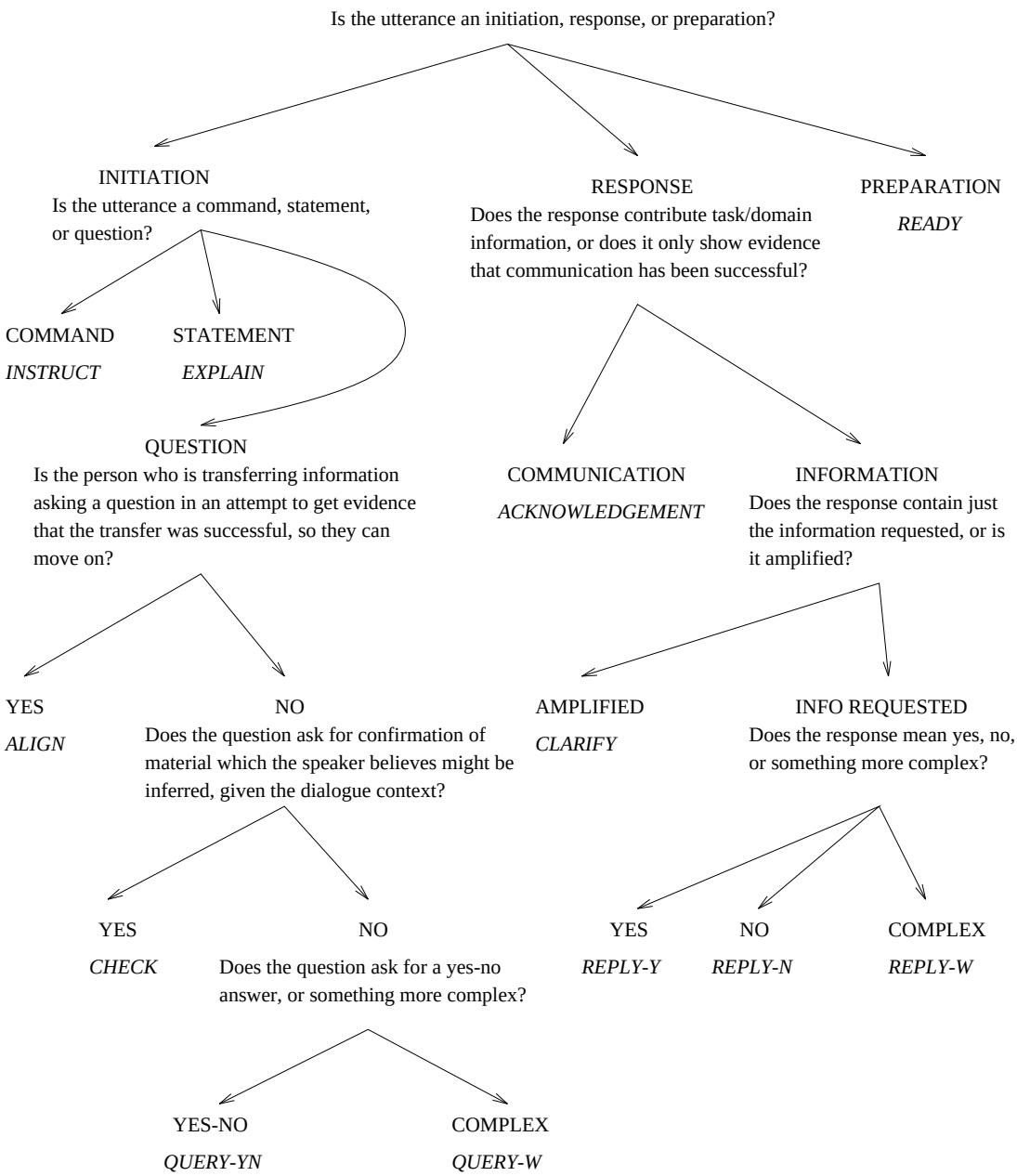


Figure 2.1: Move categorisation. Source: Carletta *et al.* (1997)

- **The *Explain* Move**

This is a declarative utterance that gives some information not elicited from the other speaker[1]. Information transferred through *explain* moves include the state of the discourse plan (such as domain information) and the establishing of mutually known information (such as shared landmarks). An example of an *explain* move is:

- *Giver*: the farm land is about three fingers east of the dead tree (*explain*)
- *Follower*: okay (*acknowledge*)

- **The *Check* Move**

Check moves are interrogatives that request the other participant to confirm some information or state of discourse goal. This information is either explicitly conveyed by the partner or inferable. Most of the *check* moves involve comparing possible matching landmarks on the map. Examples of a *check* move are given in lines 5 and 12 in table 2.1.

- **The *Align* Move**

This move is discourse function orientated in that it is used to check the agreement, attention or readiness of a partner. *Align* is mostly used to check the successful transferral of information in order to close a game and move onto the next. This is commonly realised as "Okay?" immediately after a question. It is also often used to break up larger pauses, making sure the other participant is still following the conversation. For example:

- *Giver*: so you've circled the farm land? (*align*)
- *Follower*: yes (*reply-y*)

[1]If it is elicited then the move would be a response, for example a *reply* to a *query-w*.

- ***The** Query-yn **Move***

Classification of yes/no questions is based primarily on syntactic form mostly starting with "do you have" and referring to the landmarks on the map. They are only classified as such if they take a "yes" or "no" answer and do not fall into the *check* or *align* categories. Examples of *query-yn* moves are given on lines 4 and 7 in table 2.1. The data given are dialogue initial and illustrate the frequent use of *query-yn* to establish the features on the maps.

- ***The** Query-w **Move***

This category mostly includes wh-questions (those that start with "which, what, where etc."). However, it is also a general category for all interrogatives that do not fall into the other initiating move categories, including asking the hearer to choose an answer from a list of possibilities. An example of a *query-w* move is given below.

- *Follower*: which side of the well do I pass? (*query-w*)

- *Giver*: you go around the southwest side of the well (*reply-w*)

2.4.3 Response Moves

- ***The** Acknowledge **Move***

Acknowledge moves are generally short utterances frequently used as *backchannels*. They indicate that the speaker has heard and/or understood the corresponding utterance from the other speaker. *Acknowledges* involve either producing a verbal acknowledgement (usually "okay"), paraphrasing or repeating all or part of the utterance, illustrated in lines 9 and 14 in table 2.1.

- ***The** Reply-y **Move***

This is a positive reply elicited from a question, normally a *query-yn, check* or *align*. Examples can be found on lines 3 and 13 in table 2.1.

- ***The** Reply-n **Move***

This is a negative reply elicited from a *query-yn*, exemplified on line 6 in table 2.1.

- ***The** Reply-w **Move***

This move type encompasses any type of response to a query that is not included in the previous two categories. See the section on *query-w* moves for an example.

- ***The** Clarify **Move***

A *clarify* move is a reply that contains more information than was requested by the question. This move type is used if the change in meaning is too small to merit one of the other move types. These tend to occur when the participants are not talking about a particular topic (such as a landmark), but when a general problem has occurred. For example, if the follower is unclear where he should be on the map, as illustrated in the following data extract.

- *Follower*: so I'm staying on the east side of the babbling brook and going south on it? (*check*)

- *Giver*: yeah (*reply-y*)

- *Follower*: all right (*acknowledge*)

- *Giver*: just follow it down until you get past the curve and then stop (*clarify*)

- ***The** Ready **Move***

The *ready* move is in addition to the initiating and non-initiating moves. It frequently occurs to conclude that the current game has ended and to set up the next one. Although *ready* moves basically play the role of discourse markers, it is important that they have their own class to distinguish them from *acknowledges* which are also short and have similar wording (mostly "okay", "right"). There are two examples of *ready* moves starting games given in table 2.1.

2.4.4 Games

As discussed previously, the idea of dividing discourse into games was initially developed by Power (1979) for computer-computer interactions. A game is taken as a group of moves that establish some higher level goal. A game is only complete if both participants agree that this aim has been achieved. Games are named depending on their discourse function; this is taken as the same as the initiating move, excluding *ready* moves.

Table 2.1 gives examples of four consecutive games. Take game 1, for example, this is a *query-yn* game as the first initiating move is a *query-yn* move. This game has the goal of establishing the landmark "sandy shore". Another example is game 4, which is an *instruct* game and has the goal of getting the follower to perform the action of moving down the map.

Games can be embedded where smaller subgoals need to be achieved before higher goals. Examples of embedded games are found in games 2 and 4 in table 2.1. These embedded games are *check* games that need to be completed before the goal of the main game (e.g. instructing) can be obtained.

2.4.5 Transactions

Transactions are larger chunks of dialogue that accomplish a higher level goal in

the Map Task, such as mapping out a section of the route. Labelling of transactions is not always straightforward, as participants like to review previous parts of the route and also look forward to upcoming parts to provide context. Conversation not pertaining to the route also occurs, for example in discussing the experimental makeup.

2.5 DAMSL

The game based systems described above were judged inappropriate for the complex shallow analysis needed for dialogue systems such as TRAINS (Allen *et al.*, 1996). A multi-layer dialogue annotation system was developed called DAMSL (Dialog Act Markup in Several Layers) described in Core & Allen (1997). In this scheme, each utterance is attached with a number of independent labels each pertaining to a different action. For example, an utterance may simultaneously be responding to a question, informing and promising to perform an action. DAMSL consists of a set of 33 dialogue act types. These are subdivided into three categories:

- forward communicative functions
- backward communicative functions
- utterance features

Forward communicative functions affect the future conversation, such as a request for information. Utterances that are included in this category potentially induce further action. For example: statements; offers, commitments (*committing-speaker-future-action*); information request, imperatives (*influencing-addressee-further-action*); performatives. As described in section 2.3, performatives, first defined by Austin (1962), are utterances that *perform* some action and can be

used in conjunction with any one of the other categories. For example, "You are fired" is a statement and a performative.

The *backward communicative functions* include utterances of agreement, understanding and answers. The final category is *utterance features* which tries to capture the content of an utterance. These include utterances on a meta-level (such as describing the task at hand), abandoned utterances and conventional utterances such as "hello" and exclamatory ones such as "wow".

The advantage of the DAMSL scheme is that an utterance can be assigned more than one category, as the layers of utterance types are independent to a certain extent. The drawback with this approach is that it is far more complicated than the plan based schemes and therefore has lower inter-annotator agreement. The agreement results of DAMSL and the other schemes are compared in section 2.11.

The scheme was primarily designed for task orientated dialogue systems, such as TRAINS described in section 1.5.1. Allen *et al.* (1996) do state that it is adaptable for non-task orientated dialogues such as the switchboard corpus, described in the next section.

2.6 Switchboard Data Recognition

A large corpus of spontaneous telephone conversation has been hand-annotated for dialogue structure. These data, known as the switchboard corpus (SWB), are a set of recordings of two participant conversations on a given topic. The participants do not know each other prior to the recording.

Studies on the switchboard corpus (Shriberg *et al.*, 1998; Jurafsky *et al.*, 1997) use dialogue information in an attempt to improve word recognition accuracy (see section 7.1). The dialogue act tag set they use is a modified version of DAMSL. As DAMSL is a task oriented annotation scheme, modifications were necessary

Tag	**Example**	**%**
Statement-non-opinion	*Me, I'm in the legal department*	36%
Acknowledge (Backchannel)	*Uh-huh*	19%
Statement-opinion	*I think it's great*	13%
Agree/Accept	*That's exactly it*	5%
Abandoned or Turn-Exit	*So -*	5%

Table 2.2: Examples of the five most frequent dialogue acts after clustering, with the percentage of the data they account for

for less goal orientated dialogue. The new set consists of approximately 60 basic tags some of which can be combined, resulting in a set of 220 combinations, 130 of which occurred less that 10 times. This set was clustered by hand into 42 categories. The top five most frequent of these are given in table 2.2, taken from Shriberg *et al.* (1998). Further clustering was performed resulting in main seven tags: *statements, questions, incomplete utterances, backchannels, agreements, appreciation* and *other*.

80% of the 42 SWB labels can be mapped onto the standard DAMSL labels. The main difference is that some of the SWB tags incorporate both forward and backward communications. For example, interrogatives are placed in the same class if they are eliciting information (forward communicative function) or querying given information (backward communicative function).

Shriberg *et al.* (1998) claim that the SWB annotation scheme "incorporates both traditional sociolinguistic and discourse-theoretic rhetorical relations (adjacency pairs) as well as some more-form-based labels". By using a more shallow tag set, they claim that they can cover a larger data set. This may be true but higher level discourse information, such as whether a local goal is achieved, is shown to be useful in the study presented here (see chapter 8).

2.6.1 Clarity

The Clarity project described in Finke *et al.* (1998) is a speech recognition project using similar data to the switchboard but in Spanish. In this corpus, members of the same family talk on unrestricted topics. The dialogue act annotation scheme is similar to the SWB with a few modifications. The tag set had to be changed to account for a distinction between direct and indirect speech acts, expressions of surprise and attention directives. There were also tags that could be excluded as they never occur in the data, for example reformulations of the speaker's utterance.

2.7 Verbmobil

The set of dialogue acts for the translation dialogue system described in section 1.4 consists of 43 dialogue acts described in Jekat *et al.* (1995) and Maier (1997). These are tailored towards the domain of the system, namely appointment scheduling. The 43 acts are grouped into 18 abstract illocutionary classes. The following examples are taken from Maier (1997).

- REQUEST_SUGGEST: the dialogue participant is asked to make a suggestion.
- ACCEPT, REJECT: a proposed item is accepted or rejected
- GREET, THANK, BYE: conventional dialogue actions are performed

Finer grained distinctions deal with propositional content, for example date, duration and location.

2.8 Dialogue Annotation of Classroom corpus

Sinclair & Coulthard (1975), part of the Birmingham School Discourse Analysis group, developed a set of dialogue acts in an attempt to model classroom discourse.

Their scheme consists of a hierarchy of five ranks of discourse structure: lesson, transaction, exchange, move and act. Lessons constitute a series of transactions which themselves consist of a series of exchanges. The structure of each exchange is in terms of three move types: initiation (I), response (R) and feedback (F). The IRF model predicts that these moves will occur in the given sequence and that only the initiation (I) is obligatory. Sinclair & Coulthard (1975) define 22 acts, many of which are limited to particular moves. For example, "accept" or "evaluate" are typically response type moves. The disadvantage of this scheme is that it is very domain specific. In addition, the IRF model is not very predictive as it does not specify the conditions under which the R and F type moves would be omitted.

2.9 Annotation for a Japanese Dialogue System

Nagata & Morimoto (1993) and Kita *et al.* (1996) define a classification system based on the utterance's syntax, modality or speaker's intention. Some examples of the 15 fold tag set are given below:

- *inform*: information giving utterances (e.g. "I've just received my registration card")
- *questionif*: yes/no questions (e.g. "Do you have any other questions")
- *suggestion*: speaker suggests that the hearer perform some action (e.g. "How about seeing that?")
- *acknowledge*: speaker acknowledges the other person's utterance (e.g. "I see")

The data they use are typed dialogues between a secretary and a questioner in Japanese. Their dialogue annotation scheme is similar to DAMSL in that it has many tags that can be simultaneously attached to one utterance.

2.10 Topic Structure Identification

Prosodic cues to topic shift structure are examined in Nakajima & Allen (1993) for incorporation into the TRAINS system described in section 1.5.1. Utterance types are divided into four main categories which reflect the topic state given below, taken from Nakajima & Allen (1993).

- **Topic Shift**
  - **New Topic**: utterance that introduces a new topic, new (sub)goal or new (sub)plan
  - **Topic Development:** utterance develops previous topic with some weak linkage between them
  - **Interruption:** previous or simultaneous utterance is interrupted by current utterance
- **Topic Continuation:** talking about the same plan or entity as in the previous utterance
- **Elaboration Class**
  - **Elaboration:** the current utterance adds some information relevant to the previous statement
  - **Clarification:** the current utterance clarifies some proposition from the previous statement
  - **Summary:** the current utterance summarises contents of previous utterances
- **Speech Act Continuation:** single speech act continues over several utterances

In their paper, Nakajima and Allen show that various prosodic features can be used to discriminate the above topic boundary classes. For example, they found that the mean F0 onset of an utterance containing a new topic is higher than one that continues the same topic (125Hz compared to 101Hz). Another trend they found is that the average final F0 of declarative utterances is lower if a new topic is introduced in the next utterance (88Hz compared to 108Hz). This utterance classification is beneficial from a pragmatic viewpoint and would be useful in conjunction with key word spotting to identify the current topic. However, these dialogue act types do not group utterances that are syntactically similar. In addition, the prosodic features they present to distinguish utterance types exclude interrogative features. This suggests that further investigation is needed to create a set of topic related utterance types distinguishable by both their intonation and their syntactic structure.

2.11 Assessing Discourse Annotation Schemes

When doing any type of research, it is essential that other people besides the author can understand the reasoning behind it and replicate the workings. This is especially true of annotation schemes, as one desires a corpus of consistently annotated dialogue acts across a number of coders. This section presents several methods of measuring the reliability of coders for a given annotation scheme.

Krippendorff (1980) defines three terms of coder reliability: *stability, reproducibility* and *accuracy*. *Stability* refers to whether a coder's judgement varies over time. *Reproducibility* or inter-coder variance requires different annotators to code in the same way. The final term is *accuracy* where the coder has to conform to some known standard.

Carletta (1996) and Carletta *et al.* (1997) propose using a figure that takes into account how different the results are the from random assignment of cate-

gories. The statistic that they use is the *kappa* coefficient. This figure measures pairwise agreement among coders making categorical judgements and is corrected for chance.

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.1}$$

$P(A)$ is the proportion of times the coders agree and $P(E)$ is the proportion of agreement one would expect by chance[2]. This provides a scale of accuracy from 0 to 1, where if there is no more agreement than chance, K is 0. If there is total agreement K is 1.

Krippendorff defines a threshold of $K > .8$ for good reliability and $.67 > K > .8$ for tentative conclusion to be drawn. This statistic can be used for both classification of utterances into a set of mutually exclusive classes and also to test the regularity of utterance boundary placement. These are obviously interrelated: as Krippendorff highlights, there is not much point having a high accuracy of classification if the segmentation process is inaccurate.

2.11.1 Annotation Accuracy of Game Analysis Theory

Carletta *et al.* (1997) use the kappa statistic to examine if Conversational Game Analysis theory can be applied with sufficient accuracy. In this study, the authors compare four reasonably experienced coders who each code the same four dialogues using both text and speech.

- ***Reliability of Move Segmentation***

Two figures are given for move segmentation. The first is the kappa score which is used to determine the agreement for assigning or not assigning move boundaries at each word boundary. This score is $K = .92$.

[2]See Siegel & Castellan (1988) for complete calculation instructions.

The second figure is the *pairwise percentage agrreement* which just examines agreement on word boundaries where any coder has marked a boundary. The number of pairs of coders that agree is divided by the total number of coder pairs. Pairwise percentage agreement is reasonably good for move segmentation (89%). Most of the disagreements involve the *ready* and *reply* moves. The problem with *ready* is that some coders include the following extra utterance such as an *explain* in the *ready* move while others label two separate moves. Similarly with *reply* moves: some coders make them short, with an extra move if some elaboration follows, (e.g. an *explain, clarify* or *instruct*) while others include these moves in the *reply* move.

- ***Reliability of Move Classification***

The kappa coefficient is calculated for utterances where coders agree on the boundaries. This figure is $K = .83$, which is above the threshold set by Krippendorff (1980) for good reliability. Confusions arise between the following moves: *check* and *query-yn*; *instruct* and *clarify*; *acknowledge, ready* and *reply-y*.

Separate tests were performed dividing the data into sets of initiating and non-initiating moves. For initiating moves, agreement is $K = .95$ between the following categories: statements (i.e. *explain*), commands (i.e. *instruct*) and questions. For non-initiating moves, the separate categories were between *acknowledge*, *clarify* and *replies*. This kappa score is $K = .86$. This results in an overall kappa score of $K = .89$.

- ***Reliability of Game Coding***

Testing the reliability of game coding is not as straight forward as for moves. This is because, although coders may agree on when a game starts and the goal of the game, they may not always agree on where the game ends. In addition, as games can nest within each other, it is not possible to analyse game segmentation in the

same way as for moves. Carletta *et al.* (1997) observe a 70% pairwise percentage agreement on game starts. Where starts are agreed most coders also agree on game type ($K = .86$). This is due to the fact that most of the time the move type of the initiating move is taken as the game type. Where the coders agreed on the game start, they agree 65% of the time where the game ended.

2.11.2 Agreement Results for Other Annotation Schemes

- *DAMSL*

Dialogue annotation accuracy experiments were conducted using 93 dialogues of TRAINS corpus. These data consist of dialogues between humans discussing the task described in section 1.5.1. One participant is given a problem to solve such as shipping boxcars. The second participant acts as a problem solving agent. The results of inter-annotator agreement are below reliability standard set by Krippendorff but are of a usable quality ($.67 < K < .8$). See Core & Allen (1997) for further details.

- *SWB*

Annotator accuracy of the 42 SWB dialogue acts types is $K = .8$, which is highly creditable. Grouping these categories according to the main seven types yields a kappa score of $k = .85$. Recall that the SWB dialogue acts are a modification of the DAMSL set. As the SWB kappa score is significantly higher than the DAMSL score, one can infer the modifications made were justified and indeed an improvement.

- *Verbmobil*

Inter-coder agreement between two annotators is excellent for the Verbmobil dialogue act boundary detection ($K = .9$). Classification of the 18 utterance types between these coders is also good ($K = .8$). Ideally, a comparison between a larger number of coders would be desirable. Stability of one of the coders was calculated with a ten month time lapse. This figure is $K = .94$ for segmentation and $K = .84$ for classification.

- ***Summary***

The annotation reliability figures for the above systems are not directly comparable as the dialogue acts vary in number and complexity. One can gauge the accuracy to a certain extent by using Krippendorf's standard of $K > .8$ for reliable data coding. It is worth noting that the accuracy of the classification and segmentation of the Conversational Game Analysis theory adopted in this study is above this threshold.

Chapter 3

Experimental Setup

3.1 Introduction

There are two main goals of the work presented in this thesis, the first of which is the automatic classification of utterance types. This has a number of applications, such as human computer interactive systems where the system needs to know if it is being asked a question or not. Automatic utterance type classification would also facilitate the annotation of large electronic databases. Finally, the utterance type detector can be used in an automatic speech recognition system (ASR). This leads us to the second goal of this work, which is to be able to perform utterance type recognition with enough accuracy to improve word recognition error in an ASR system.

This chapter is divided into two parts. The first part gives an overview of the type of data used in the experiments and the choice of dialogue annotation scheme. The second part gives an overview of the method used to identify utterance type and how this is used in an automatic speech recognition system to improve word error rate.

3.2 The Data

The data used in the experiments described in this thesis are a subset of the DCIEM Map Task corpus. This is a corpus of spontaneous goal-directed dialogue speech produced by Canadian males (Bard *et al.*, 1995)[1]. This corpus was chosen over a Map Task corpus produced by Glaswegian students in order to take advantage of the large amount of work on North American speech recognition. In addition, it was thought that the Canadian intonation would be easier to model than the Glaswegian, which is generally more monotone.

3.2.1 The Map Task Scenario

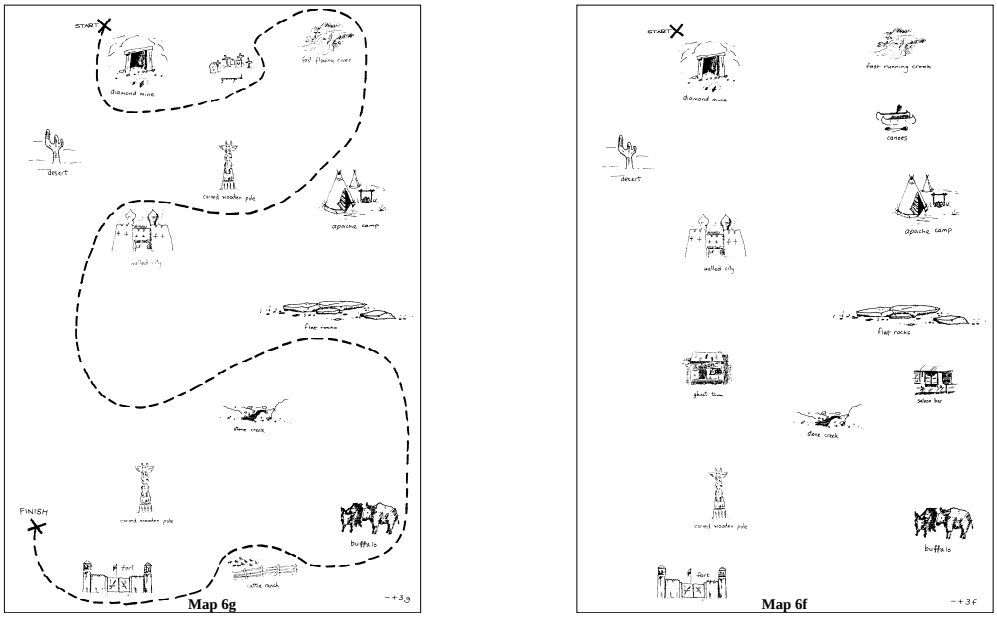
In the Map Task scenario (Bard *et al.*, 1995), each conversation has two participants with different roles called *giver* and *follower*. Each participant has a map with landmarks. These maps are similar to each other but not identical as illustrated in figure 3.1. The role of the *giver* is to guide the *follower* through the route on his map.

The speakers are recorded using separate high-quality microphones. Although there is some speaker overlap, one can basically distinguish different speakers by their channels.

3.3 Choice of Discourse Annotation Scheme

A number of possible dialogue act annotation schemes have been discussed in chapter 2. These schemes take two basic approaches: *shallow* discourse structure or a *deep* structure. The shallow level dialogue labels are adopted for complex dialogue systems, e.g. TRAINS (Allen *et al.*, 1996) and less task specific speech

[1]Although these data are collected from a sleep deprivation experiment, dialogues recorded in non-standard conditions are not included in the data



(a) Giver (b) Follower

Figure 3.1: Example of two slightly different maps given to the Map Task participants

recognition applications such as SWB (Shriberg *et al.*, 1998). These schemes, however, do not model the hierarchical nature of discourse.

Deep structure schemes include plan based schemes such as that developed by Power (1979). His system specifies discourse goals and how to achieve these goals using set plans. He developed this scheme for the computer-computer scenario where the agents have to perform a simple task, such as opening a door and moving through it. The Map Task is similar to this scenario in that the two participants have to cooperate to achieve the discourse goal of moving from A to B on the map. Therefore, Carletta *et al.* (1997) modified his scheme for the Map Task dialogues, as described in section 2.4. As discussed in section 2.11.1, Carletta shows that the Conversational Game Analysis theory can be applied with a high degree of reliability.

Data set	Number of dialogues	Number of utterances	Purpose
Set A	50	12,758	training
Set B	20	3726	training
Set C	5	1061	testing

Table 3.1: Statistics for training and testing sets of the DCIEM corpus

	Word	Intonation	Moves	Games
Set A	yes	no	yes	no
Set B	yes	yes	yes	yes
Set C	yes	yes	yes	yes

Table 3.2: Hand-labelled training and testing sets of the DCIEM corpus

3.3.1 Data Annotation

The data is divided into two sets, one for training (A) and one for testing the system (C). None of the test set speakers are in the training set, i.e. the system is speaker independent. Set B is a subset of a larger training set A. Details of these sets are given in tables 3.1 and 3.2.

Word transcriptions are available for all the data. These, however, are not aligned in time. Intonation events are hand-labelled for the smaller training set B and the test set C. All the data are hand-labelled for *moves*, but the larger discourse unit of *game* is only labelled for sets B and C.

- ***Frequency of Move Types***

Table 3.3 shows the frequency of the different moves in the training set divided between the follower and the giver. One can see how the nature of the Map Task affects the discourse structure with the giver uttering many *instruct* and *query-yn* moves and the follower many *acknowledges*. The high number of *explain* moves by the follower may be surprising. These are often uttered in an attempt to establish the different landmarks on the maps. The *acknowledge* move has a high frequency, thus illustrating the number of backchannels in discourse. Another

Move type	Frequency	Giver frequency	Follower frequency
instruct	604	596	8
explain	330	148	182
align	120	117	3
check	245	62	183
query-yn	333	246	87
query-w	97	24	73
acknowledge	922	292	630
clarify	93	85	8
reply-y	384	174	210
reply-n	107	27	80
reply-w	145	71	74
ready	346	289	57

Table 3.3: Frequency of move types for set B training set, follower's moves and giver's moves

factor contributing to the number of *acknowledges* is the fact that the Map Task is not done face-to-face. A study of the Glasgow Map Task (Anderson *et al.*, 1991), which is coded for eye contact, shows that the task is completed in less moves and 13% fewer words when eye contact is present. This is due to the fact that many of the acknowledgements are expressed non-verbally.

The difficulty of the task of move type detection is reflected in the distribution of the moves to a certain extent. One defines a figure of *chance* that is the percentage of moves that would be correctly classified if the most frequent move type was picked 100% of the time. For this database chance is 24%, which is the proportion of *acknowledge* moves. Another measure of task difficulty is *perplexity* which is defined in chapter 4.

3.4 System Architecture

This section is divided into two parts. The first deals mainly with the method of automatic move classification. The second part looks at how this can be integrated into the automatic speech recognition system.

3.4.1 Automatic Move Classification

There are three basic sources of information that can be tapped for move recognition. Firstly, moves of certain types follow each other with a degree of regularity. For example, as the data in table 2.1 illustrate, *query-yn* moves are often followed by *replies* which in turn may be followed by an *acknowledge*. This sequence of moves is more likely than, for example, three *acknowledges*. The regularities in move sequence are captured by a *dialogue model* (DM). This dialogue model takes advantage of the fact that one can tell the role of the speaker automatically. This is very useful as the different participant roles have different distribution of moves, as discussed above.

Secondly, utterances of a certain move type have similar syntactic and lexical patterns. For example, a *query-yn* frequently starts with the words "Do you have a ...". A language model (LM) is used to capture these characteristics. One language model is trained for each of the move types. The recogniser is run 12 times using a different language model each time. Whichever language model matches the string of recognised words the best is taken as the most likely move type.

Finally, intonation is indicative of move type. For example a *query-yn* frequently has a rising intonation contour, while an *explain* frequently has a falling one. The intonation model (IM) is used to produce the likelihood of the intonation given the different move types.

Figure 3.2 gives a schematic representation of how these three models are combined to perform move recognition. The architecture of the system is the same as that reported in Taylor *et al.* (1998b) and King (1998)[2]. Finding the most likely move sequence is a search problem. The search space is a list of all the possible combinations of move types for a given number of utterances. The solution to

[2]These describe a joint project with the author.

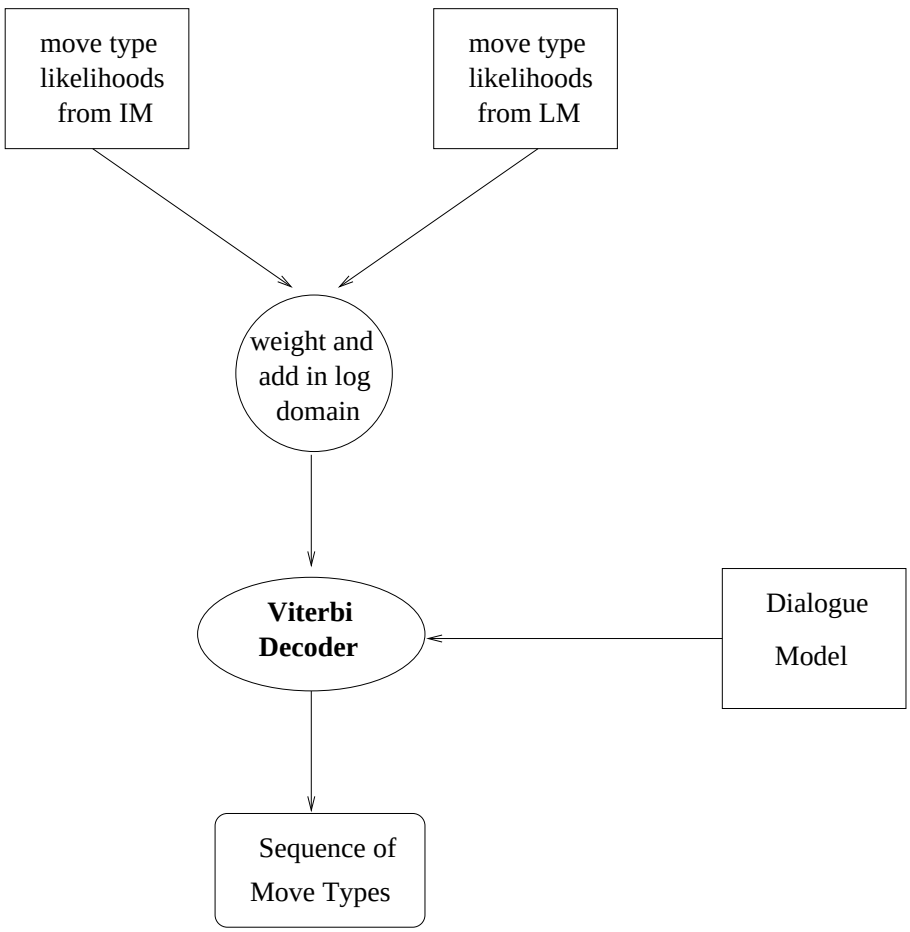


Figure 3.2: System Architecture

this problem is finding the most likely sequence based on observations about each utterance combined with prior knowledge. The observations are provided by the intonation and language models. The prior knowledge is provided by the dialogue model that looks at the move type of previous utterances.

To find the best move sequence, one could perform a brute force search through all the possible move sequences. As this is computationally expensive, the mathematical algorithm known as the *viterbi search* (Viterbi, 1967) has been chosen. This procedure cuts down the search space as it searches incrementally, keeping track of a number of the best hypotheses in parallel. This set of hypotheses can be large and therefore would need to be pruned.

The search finds the best sequence of moves $M\star$ using the Bayesian equation 3.1. A formal derivation, taken from Taylor *et al.* (1998b), is given below:

D	the dialogue
C	cepstral observations for D
I	intonation observations, such as F_0
M	the sequence of move types for D
S	the sequence of speaker identities for D

$$M^\star = \max_M \underbrace{P(M|S)}_{\text{dialogue model}} \cdot \underbrace{P(C|M)}_{\text{speech recogniser}} \cdot \underbrace{P(I|M)}_{\text{intonation model}} \tag{3.1}$$

- ***Formal Derivation***

Given a dialogue D, the goal is to find the most likely move sequence ($M\star$) given the following information sources: speaker identity (S); acoustic or cepstral

observations (C); and intonation features (I).

$$\begin{aligned} M^\star &= \operatorname*{argmax}_M P(M|S,C,I) \\ &= \operatorname*{argmax}_M P(M)P(S,C,I|M) \end{aligned}$$

Assuming that S, C and I are independent:

$$\begin{aligned} &= \operatorname*{argmax}_M P(M)P(S|M)P(C|M)P(I|M) \\ &= \operatorname*{argmax}_M P(S)P(M|S)P(C|M)P(I|M) \end{aligned}$$

and since $P(S)$ is a constant for any given D:

$$= \operatorname*{argmax}_M P(M|S)P(C|M)P(I|M) \tag{3.2}$$

In this derivation, one assumes that speaker identity has no effect on the acoustic or intonation features. Although this is clearly false, one already makes this assumption by using the same intonation and word recogniser for both participants. The following chapter shows that the likelihoods from the dialogue model are calculated without regard to intonation and acoustics.

The middle term in equation 3.1 is the output likelihood of the acoustic phonetic recogniser. Specifically, the likelihood of the cepstral coefficients is calculated from the recogniser by using 12 different language models, as shown in the following equations.

Letting W range over all possible word sequences,

$$\begin{aligned} P(C|M) &= \sum_W P(C|W)P(W|M) \\ &\approx \max_W P(C|W)P(W|M) \end{aligned} \tag{3.3}$$

The sum over all possible word sequences is replaced by the most likely sequence, i.e. the output of the recogniser. Stolke *et al.* (1998) suggest using a lattice of the N-best word sequences from the recogniser to calculate the likelihood for each move type. This method may improve the results given in chapter 7 but it is very computationally expensive.

Let

$$\begin{aligned} c_i &= \text{cepstral observations for the } i\text{th utterance} \\ C &\equiv \{c_1, c_2, \ldots c_{N_U}\} \\ W_i &= \text{the word sequence for the } i\text{th utterance} \\ \mathbf{W} &= \{W_1, W_2, \ldots W_{N_U}\} \\ m_i &= \text{move type of the } i\text{th utterance} \\ M &\equiv \{m_1, m_2, \ldots m_{N_U}\} \end{aligned}$$

The two terms in equation 3.3 are

$$P(C|\mathbf{W}) = \prod_{i=1}^{N_U} P(c_i|W_i)$$

which is given by the HMMs in the speech recogniser, and

$$P(\mathbf{W}|M) = \prod_{i=1}^{N_U} P(W_i|m_i)$$

which is given by the move specific language models.

3.4.2 Automatic Speech Recognition

The motivation behind using an automatic move detection system in a speech recognition system is to make it easier to choose among a number of word possibilities. In other words, one wants to take advantage of the fact that certain words and word sequences occur in one move type. For example, most *acknowledge* moves contain "okay". In chapter 4, it will be shown that the difficulty of the task of word identification is reduced by using language models specific to the utterance's move type. The method given in the previous section is used to identify the most likely move type of an utterance. This is then used to determine the language model to be used during recognition. Chapter 7 gives the word recognition results and shows that the baseline system can be improved using this novel technique.

3.4.3 The Baseline System

The baseline speech recognition system that provides the likelihoods of word sequences which are used to predict the utterance type is a standard HMM based system built using the HTK toolkit (Young *et al.*, 1996). This system uses the standard features for training the HMM models, namely 12 cepstral co-efficients plus energy, plus their first and second derivatives giving 39 component observation vectors. The HMM models are 8-component Gaussian mixture tied-state cross-word triphone models; see Young *et al.* (1996) and Rabiner & Juang (1994) for more details.

Approximately three hours and twenty minutes of speech were used to train the models. These data have a vocabulary of around 900 words, which is not particularly large in speech recognition terms. Using a general bigram language model achieves a word error rate of 24.8%[3]. This is the baseline result that one

[3]For further detail of the baseline system see King (1998).

is trying to improve by using move-specific language models.

3.4.4 Measuring Success

Move recognition results are given in terms of percentage of utterances correctly classified. Word recognition is given in terms of percentage correct and accuracy. Accuracy is used to account for words inserted or deleted and is calculated by subtracting the percentage of insertions and deletions from the percentage correct.

3.4.5 Testing Scenarios

There are three different types of scenarios where this system can be tested. The first is called *overhearer*, where the recogniser's goal is to transcribe both participants' moves and words. In this case, the predictors, excluding speaker identity, have to be guessed by the recogniser. This scenario is used for most of the experiments described in this thesis and in Taylor *et al.* (1998b). The second application is the *participant* scenario, where the computer knows what one participant is saying and can use this to predict what the other person is saying. The final *transcription* scenario uses the hand-transcribed data to predict the move or word sequence. This is the easiest of the three tasks. This scenario is adopted in the experiments reported in Poesio & Mikheev (1998) described in chapter 8.

Chapter 4

Dialogue and Language Models

4.1 Introduction

It is everyday knowledge that words do not follow each other randomly. For example, a determiner followed by a noun followed by a verb is more likely than three determiners in a row. One can take advantage of this in speech recognition to make it easier to choose among a number of word possibilities. In order to do this, statistical language models (LM) are trained on sequences of words. A language model gives the probability of a particular word given a number of preceding words or part of speech categories.

In a similar way, utterances of different types follow each other in a particularly predictable way. For example, a reply is likely to follow a question. Language modelling techniques are applied in order to model these sequences of utterance types or moves. These models are referred to as discourse grammars or dialogue models (DM).

In the first section, a standard language modelling technique, known as the N-gram model is discussed. The method of training this model is described and various techniques to cope with sparse data problems are discussed[1]. In order

[1]For a more comprehensive review of language modelling techniques see the CMU toolkit manual (Rosenfeld & Clarkson, 1997).

to examine whether the language and dialogue models improve the predictability of the data, *perplexity* and *entropy* are calculated. Section 4.2.4 discusses the significance of these figures and describes how to calculate them.

Section 4.4 looks at how the language modelling techniques can be applied to dialogue modelling. As dialogue modelling is more relevant to the topic of this thesis, a more detailed review of previous literature will be presented along with a summary of other systems that attempt to model dialogue in a similar way. Experiments taken from King (1998) show the effects of using various types of predictors in terms of move sequence perplexity for the DCIEM Map Task corpus. These results are given in 4.5.

Move-specific language models are used in move recognition. Each language model is run over the recognised word sequence and the likelihood that the utterance is of a certain type is calculated. Section 4.6 presents these move recognition results in conjunction with the best dialogue model.

The work on language modelling and dialogue modelling reported in this chapter is mostly taken from King (1998) but is part of a group project by Taylor, Isard, King and Wright reported in Taylor *et al.* (1998b).

4.2 Language Modelling Techniques

4.2.1 N-grams

The N-gram model is a statistical method that improves the identification of the current word by taking into account the a priori probabilities of various candidate identities. This method is easily trainable and can be integrated into the probabilistic framework described in chapter 3. As discussed in the introduction N-grams can also be used to model sequences of moves. For simplicity, N-grams will be described in terms of words in the following discussion. Section 4.4 deals with N-grams for dialogue modelling.

If W is a sequence of words w of length Q:

$$W = w_1, w_2, ...w_Q \tag{4.1}$$

calculating the probability of W is shown in equation 4.2. The equations below are taken from Rabiner & Juang (1993)

:

$$\begin{aligned} P(W) &= P(w_1, w_2, ...w_Q) \\ &= P(w_1), P(w_2|w_1)...P(w_Q|w_1 w_2...w_{Q-1}) \end{aligned} \tag{4.2}$$

Unfortunately, it is not practically possible to calculate the probability of a whole sequence of words of any length Q. Hence the N-gram was developed that looks $N-1$ tokens left of each word to be predicted, w_i

$$P(w_i|w_1 w_2...w_{i-1}) \approx P(w_i|w_{i-N+1}...w_{i-1}) \tag{4.3}$$

The probability of a sequence of words W is defined by the N-gram language model $P_N(W)$. This is the product of the probabilities for each w_i in Q given its context, i.e.

$$P_N(W) = \prod_{i=1}^{Q} P(w_i|w_{i-1}, ..., w_{i-N+1}) \tag{4.4}$$

The conditional probabilities are calculated using a simple frequency count shown in equation 4.5.

$$P(w_i|w_{i-1}, ..., w_{i-N+1}) = \frac{C(w_i, w_{i-1}, ..., w_{i-N+1})}{C(w_{i-1}, ..., w_{i-N+1})} \tag{4.5}$$

4.2.2 Backing Off

Ideally N would be as high as possible in order to take advantage of as many predictors as possible. However, if N is too large then this will increase the number of occurrences of N-grams in the testing set that do not occur in the training set. In order to compensate for this, a process known as *backing off* can be employed (Katz, 1987). In order to give these rare N-grams some non-zero probability, some of the frequencies from the other N-grams are *discounted*, so that the sum of the likelihoods remains equal to 1.

For example, one may want to calculate the probability of a trigram, $P(a, b, c)$. If one has not seen this sequence in the training data frequently enough to reliably estimate the probability, one can use the probability of the bigram $P(c|b)$ to estimate the trigram probability.

Let

$$P(c|a,b) = \alpha.P(c|b) \text{ if } C(a,b,c) \text{ is below some threshold} \quad (4.6)$$

where α is the backing off weight which is needed to ensure that the probabilities of the word history (a, b) sum to one (taken from King (1998)).

$$\sum_{w \in V} P(w|a,b) = 1 \ \forall \ a,b \quad \text{where } V \text{ is the vocabulary } a,b,c\ldots \quad (4.7)$$

For α to be greater than zero one needs to discount the probability mass from N-gram frequencies that are greater than zero. The following equations are taken from Katz (1987).

$$P(w_N|w_1^{N-1}) = \begin{cases} \tilde{P}(w_N|w_1^{N-1}) & \text{if } C(w_1^N) > k \\ \alpha_{w_1}^{w_{N-1}} \cdot \tilde{P}(w_N|w_2^{N-1}) & \text{otherwise} \end{cases}$$

where w_1^N is the sequence of words $w_1, w_2, \ldots w_N$, k is a threshold and $P(\cdot)$ is now estimated $\tilde{P}(\cdot)$ using :

$$\tilde{P}(w_N|w_1^{N-1}) = \frac{C(w_1, w_2, \ldots w_N) - d(C(w_1, w_2, \ldots w_N))}{C(w_1, w_2, \ldots w_{N-1})} \quad (4.8)$$

The discounting function $d(\cdot)$ allows the distribution of probability from higher frequency n-grams to lower frequency ones. There are a number of discounting functions available such as fixed and linear discounting (Ney *et al.*, 1994), Good Turing Method (Church & Gale, 1991) and Witten Bell discounting function (Witten & Bell, 1991). A detailed discussion of these methods is beyond the scope of this thesis.

One simple solution to the sparse data problem is to include a floor value (F) for the frequency of any given N-gram:

$$P(w_N|w_1^{N-1}) = \begin{cases} P(w_N|w_1^{N-1}) & \text{if } C(w_1^N) > F \\ F & \text{otherwise} \end{cases} \quad (4.9)$$

4.2.3 Interpolation Techniques

An alternative way to deal with cases where $C(w_{i-1}, ..., w_{i-N+1}) < k$ is a method known as smoothing. For example, trigrams can be smoothed with bigrams and unigrams. In order to deal with combinations that do not occur in the training set (but may occur in the test set), one can use a weighted sum of the frequencies from three models, as shown in equation 4.10, taken from Rabiner & Juang (1993).

$$\tilde{P}(w_3|w_1, w_2) = p_1 \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)} + p_2 \frac{C(w_1, w_2)}{C(w_1)} + p_3 \frac{C(w_1)}{\sum C(w_i)} \quad (4.10)$$

where $p_1 + p_2 + p_3 = 1$ and $\sum C(w_i)$ is the size of the corpus.

The interpolation weights (p_x) are calculated using Estimation Maximisation

(EM). This algorithm is implemented in the CMU toolkit (Rosenfeld & Clarkson, 1997) which is used by King to train the language models discussed in section 4.3.

4.2.4 Perplexity and Entropy

This section describes how to calculate a measure of how well a language or dialogue model reduces the difficulty of the task of word or move recognition. Data complexity is calculated by measuring *entropy* and *perplexity*. Perplexity is often known as the average word branching factor. In other words, given a sequence of X words, how many possible words are there that could follow? The better the language model, the more constraints there are, the lower the perplexity and the easier the recognition task. Suppose the data contains a vocabulary of X different words which occur in equal quantities, the perplexity would be X because for each word there would be X equally likely possible words that could follow. If one word is more likely than another then the perplexity would be less that X.

Accuracy of recognition is not directly related to perplexity, for example, words that are badly recognised may be predicted the most often by the language model. However, in general, if a language model lowers the perplexity of the source then one expects a better word accuracy. The perplexity and entropy of the data can only be estimated from the test set. When performing any optimisation method, such as calculating interpolation weights, perplexity should be calculated using a held-out data set.

The equations below are taken from Rabiner & Juang (1993). Equation 4.11 shows perplexity B in terms of entropy H.

$$B = 2^H \tag{4.11}$$

The entropy of the test set is calculated for a sequence of Q words by equation 4.12 where Q should be as large as possible.

$$H_p = -\frac{1}{Q} log P(w_1, w_2 ... w_Q) \tag{4.12}$$

H can be estimated by using $P(W) = P(w_1, w_2 ... w_Q)$ from the language model. For example, using the language model P_N in equation 4.4 gives the following estimate H_p

$$H_p = -\frac{1}{Q} log \tilde{P}(w_1, w_2 ... w_Q) \tag{4.13}$$

$$H_p = -\frac{1}{Q} \sum_{i=1}^{Q} log P(w_i | w_{i-1}, w_{i-2} ... w_{i-N+1}) \tag{4.14}$$

H_p is the average difficulty of classifying a word based on the language model. Therefore the lower H_p, the more the language model reduces the difficulty of the task.

4.3 Language Modelling Experiments using the Map Task Data

4.3.1 Experimental Setup

The previous section discussed the methodology behind training language models to capture regularities in word sequences. The assumption behind this study is that utterances of different types have characteristic syntactic patterns and lexical distributions. For example, a *query-yn* move frequently starts with the word sequence "Do you have a ...?". Training language models that are specific to utterance types will cut down on the number of possible words and therefore reduce the likelihood of word recognition error.

Move type	Sentences	Words
acknowledge	2607	6363
align	319	1753
check	598	4359
clarify	246	2149
explain	733	6521
instruct	1407	17991
query-w	262	1863
query-yn	703	5748
ready	784	1574
reply-n	262	770
reply-w	331	2937
reply-y	1020	2824
total	9272	54852

Table 4.1: Move-specific LM training set sizes

King (1998) calculates the perplexity of the test set using a general language model and move-specific language models. Some of the moves are not as frequent as others (see table 4.1) and have fewer words to train on. King, therefore, smoothes the move specific language models with a general model. This takes advantage of the robustness of the general model while still holding onto the characteristics associated with an utterance type. This is done by combining the counts $C(\cdot)$ in the move specific and the general model using weights. This is shown in equation 4.15, taken from King (1998).

$$\begin{aligned} C_{\text{combined model}}(a,b) &= w.C_{\text{type-specific data}}(a,b) + \\ &\quad (1-w).C_{\text{all data}}(a,b) \end{aligned} \tag{4.15}$$

As discussed above, w is calculated using the EM technique. The weightings of the move-specific language models (w) are given in table 4.2. This table shows that for shorter move types the move-specific language models are more heavily weighted. This is because they have a stricter syntax and are therefore very

Move type	Weight
acknowledge	0.8
align	0.5
check	0.4
clarify	0.3
explain	0.5
instruct	0.7
query-w	0.6
query-yn	0.6
ready	0.9
reply-n	0.9
reply-w	0.4
reply-y	0.8

Table 4.2: The interpolation weights of the move-specific language models

effective in predicting word sequences. For example, *acknowledges* mostly consist of "okay" or "right". On the other hand, longer moves such as *explain* and *check* have a more varied syntax and larger variation in their lexical distribution. Therefore, the general model is heavily weighted so one can benefit from its larger vocabulary.

4.3.2 Perplexity Results

Perplexity results are calculated using the following language models trained on set A and tested on set C of the DCIEM corpus described in 3.2.

1. General language model (trained on set A)

2. Move specific language models (each trained on a subset of set A according to utterance type)

3. Smoothed models 1 and 2

4. Best choice of 1, 2 and 3 for each utterance type

Models 1, 2 and 3 have been discussed above. Method 4 simply takes the model that has the lowest perplexity for each of the move types. For example, using the general model for utterances of types: *reply-w*, *clarify* and *explain* results in the lowest perplexity. On the other hand, the smoothed models are best used to recognise words in utterances of types: *acknowledge*, *align*, *check* , *query-w* and *query-yn*. Move specific models are used for utterances of types *instruct, reply-n* and *reply-y*.

Table 4.3 is taken from King (1998) and gives the whole test set perplexities using the four different methods.

Model	**Test set perplexity**
general (baseline)	23.6
original move-specific	22.1
smoothed move-specific	21.5
best choice move-specific	21.0

Table 4.3: Language model perplexities

This shows an improvement by smoothing the original move type-specific and the general language models. The "best choice" method allows one to pick the appropriate method depending on the move type of each utterance and results in a perplexity of 21.0. This method was therefore adopted in the automatic speech recognition experiments described in chapter 3. Using these sub-language models results in a reduction in word error rate. The word recognition results are given in detail in chapter 7.

4.4 Dialogue Modelling

This section describes the dialogue model component of the system. N-gram dialogue models are used to calculate the prior probability of a move $P(M)$ in equation 3.1 on page 42.

One can adopt two approaches to dialogue modelling: top down or bottom

up. Either of these approaches can be linguistically motivated and statistically trained.

4.4.1 Top-down Approaches

- ***Stochastic context free grammars***

Stochastic context free grammars (SCFG) are top-down generative models used frequently in natural language processing. The grammar is made up of a finite set of rules. Each non-terminal node maps onto a sequence of non-terminal or terminal tokens. For example:

$$\begin{aligned} \text{instruct_game} &\longrightarrow \textit{instruct}\ \text{clarification_game}\ \textit{acknowledge} \\ &\longrightarrow \textit{instruct acknowledge} \end{aligned}$$

$$\begin{aligned} \text{clarification_game} &\longrightarrow \textit{clarify reply} \\ &\longrightarrow \textit{clarify} \end{aligned}$$

Probabilities which are derived automatically from the training data can be associated with each mapping. This rule based technique may work in small domains. However, in larger domains such as the Map Task, it is likely to result in inadequate coverage. In addition, SCFG are not compatible with the viterbi search algorithm explained in section 3.4.1 (Rabiner & Juang, 1993).

- ***Plan Based Schemes***

Top down plan based schemes, such as Power (1979) described in chapters 1 and 3, were developed to allow two computer agents to talk to each other in a limited world. In order to complete a discourse and achieve a goal, they follow *planning procedures*. These are top down prescriptive descriptions of dialogue in terms of a

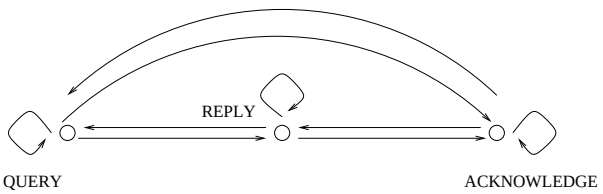


Figure 4.1: Move bigram finite state network; not all arcs are given

stack. They are, however, too simple for human-human dialogue in the real world. One particular aspect of his work that is of use is the idea that discourse can be divided into games which consist of a sequence of moves. Each game, similar to a planning procedure, has a goal and may contain sub-games which achieve necessary subgoals. However, the bottom up approach was chosen in the present study as the dialogue is described in terms of sequences of moves which make up games.

4.4.2 Bottom-up Approaches

- ***Finite State Networks***

This sequence of moves can be modelled by a finite state network (FSN). A simplified FSN is given in figure 4.1 where the nodes represent the move type and the arcs the possible transitions from one move to the next.

Bennacef *et al.* (1995) developed a non-probabilistic FSN for an air travel dialogue system. This FSN was used to aid dialogue act identification and to generate appropriate responses according to the current dialogue state. The FSN has nodes for the different stages of dialogue: opening and closing formality and information retrieval. FSNs are useful in small domains as they specify all of the occurring transitions. However, developing FSNs can be rather impractical as they have to be developed manually. They can increase in size and complexity rapidly, especially as one starts increasing the number of predictors (N).

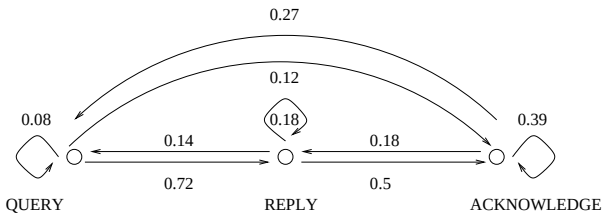


Figure 4.2: Probabilistic move bigram finite state network; not all arcs are given

- ***Hidden Markov Models***

Some of the transitions of the FSN in figure 4.1 are more frequent than others. For example, a *query* is likely to be followed by a *reply*, whereas a *query* followed by an *acknowledge* is less frequent. This is formalised by adding probabilities to the arcs of the network resulting in the probabilistic finite state network shown in figure 4.2. These show the probability of a move given the previous move, i.e. N=2.

Markov models are essentially probabilistic FSN with the addition of observation likelihoods associated with each node or state. Woszczyna & Waibel (1994) use Markov and hidden Markov models for dialogue act detection in a speech-to-speech dialogue system called JANUS. They train a Markov model with six nodes each corresponding to a type of dialogue act. The transitional probabilities are the bigram probabilities of dialogue act transitions, illustrated in figure 4.3. The observation probabilities are the likelihoods of observing a word in a given dialogue state. An improvement was found by automatically clustering words into classes depending on their context. Using this Markov model one can compute the probability of being in a certain dialogue state at any point in the discourse. Using hand transcribed word sequences, they yield a dialogue act recognition rate of 62.3%. In a second experiment, they use a hidden Markov model to cluster similar phrases into speech acts instead of explicitly linking the nodes with a speech act type. They show that this yields a reduction in test set perplexity indicating that the words have been clustered to form better language models.

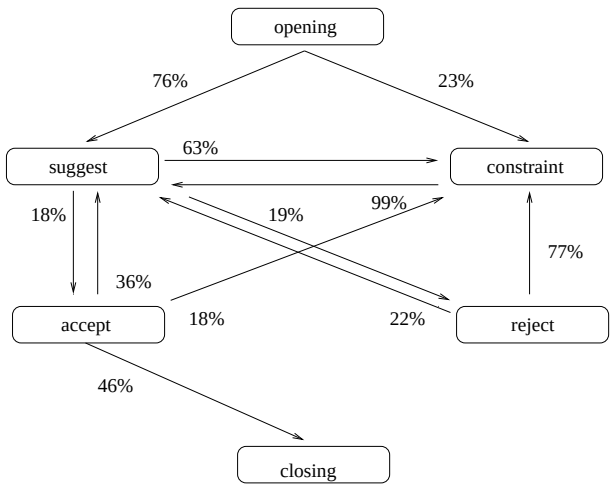


Figure 4.3: Markov model of dialogue with transitional probabilities. Source: Woszczyna and Waibel (1994)

4.4.3 N-grams for Dialogue Modelling: Previous Work

- ***Verbmobil***

Reithinger *et al.* (1996; 1997) present methods of training dialogue models using N-gram techniques for use in the Verbmobil project. Their goal is to automatically recognise the 18 acts described in chapter 2.

The dialogue model component described in Reithinger *et al.* (1996) is a basic N-gram and is improved upon using various techniques. Including speaker identity increases their baseline result from 72.2% to 75.5% [2]. In a separate experiment, they use speaker information to mirror the data, providing an alternate set of data with the speaker counterpart. That is (as far as I can tell), they train the model using a new set of moves which are the product of the move type and the speaker identity, for example **reject** is **reject-ab** or **reject-ba**. They can then collapse the new moves down to the original (e.g. **reject**) and calculate the recognition result which is 76.05%. The chance figure is 25%, which is the proportion of the most frequent dialogue act.

[2]Results are only given for hit rates within the top three answers using only the dialogue model.

Reithinger *et al.* (1997) give results for combining this dialogue model with the language model likelihoods in separate experiments for English and German data. This gives an accuracy rate of 67% for recognising the 18 dialogue acts in German and 72% for the English data. They yield better results for fixed sayings such as "great/bye" and dialogue acts such as ACCEPTS/REQUESTS. Dialogue act types that have a larger lexical distribution such as DIGRESS_SCENARIO are badly recognised.

- ***Nagata and Morimoto***

Nagata & Morimoto (1993) use a similar method of combining language model and dialogue model likelihoods. They also use hand transcribed word sequences as the input. Their goal is to predict one of 30 dialogue act types described in chapter 2. For the dialogue model component, they interpolate a trigram dialogue model with bigram and unigram probabilities with weightings of 77%, 13% and 10% respectively (see equation 4.10). With this dialogue model, they achieve a DA recognition rate of 39.7% (chance is 35%). They find that recognition of acts corresponding to the second part of an adjacency pair (e.g. *acknowledges*) is much easier than predicting initiating moves (e.g. *inform*). They also show a reduction in word perplexity by using utterance type specific language models specified by the dialogue model.

4.5 Dialogue Modelling Experiments for the Map Task

4.5.1 Simple N-grams

The first experiments to develop a statistical model of the Map Task dialogue simply use the previous N-1 moves to predict the current move. A 4-gram is illustrated in figure 4.4, using the three previous moves to predict the current

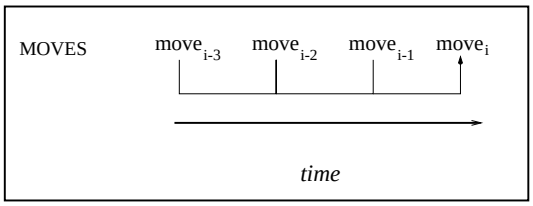


Figure 4.4: 4-gram for move sequence modelling

N	Perplexity
1	9.1
2	6.3
3	6.1
4	6.8

Table 4.4: Perplexities of simple N-gram dialogue models. Source: King (1999)

$move_i$. Table 4.4 gives the perplexity results for N-grams of increasing N. As one can see, the trigram model reduces the perplexity the most. The perplexity does not decrease proportionally to the number of predictors used, i.e. the higher N. This is because not all sequences of the predictors are likely to occur if N is greater than 3. Therefore, the dialogue model is less effective in reducing the uncertainty of the following move type.

- ***Backing Off and Interpolation***

For $N > 2$, King (1998) had to use a floor value; this compensates for N-grams that do not occur in the training set but do in the test set (see equation 4.9). Experiments were conducted by the current author to see if interpolating the trigram dialogue model with bigram and unigram models would result in a reduction in perplexity, as described in section 4.2.3. This was not the case, suggesting that using a floor is sufficient to compensate for N-grams that do not occur frequently.

4.5.2 Including Speaker Information

Dialogue models can be formed using different types of predictors. As shown in the previous section, the preceding move sequence can be used to predict the current move type. However, if the system is to be totally automatic, it has to use its own classification of the previous moves, which will not be 100% accurate. One source of information that is derivable automatically with 100% accuracy[3] is the identity of the speaker at a given point in the dialogue. As discussed in section 3.2, each participant has a different role in the Map Task. The *giver* gives instructions and directs the *follower* around the map. Therefore, each speaker has a different distribution of move types with the *giver* uttering more initiating moves and the *follower* more non-initiating. In order to capture this difference, the role of the current speaker is used as a predictor in the dialogue model.

The sequencing of the different speakers can also help to predict move types. For example, if the *giver* makes a question type move, and follows it up with a second move, one can expect the follow up to be some type of explanation, clarification or question. If it is the *follower* that utters the second move, one would expect it to be some type of reply or acknowledgement. Therefore, the identity of the previous speaker is included in the dialogue model.

A complete list of the predictors used in the experiments is given in table 4.5. The vocabulary for the predictors contains 15 elements: the 12 move types, two types for speaker identity and a value for dialogue start (ENTER). This general vocabulary for the different types of predictors may result in the creation of impossible N-grams (such as *follower* as a move type). However, this does not affect the accuracy of possible N-grams as the recogniser will ignore the impossible ones. The system has to choose from one of 13 elements in the predictee vocabulary which consists of the 12 move types and a type for dialogue end (EXIT). The

[3]If the speakers are using different microphones or phone lines.

Predictor	**Symbol**
Move type of current move	m_i
Identity of speaker of current move	s_i
Identity of speaker of previous move	s_{i-1}
Move type of previous move	m_{i-1}
Move type of other speaker's last move	m_{i-j}

Table 4.5: Notation of N-gram candidate predictors

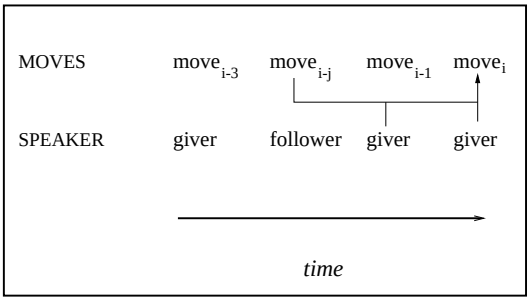


Figure 4.5: Dialogue Model III

software for training the dialogue is part of the speech tools developed by Taylor *et al.* (1998a).

4.5.3 Results

A number of predictor combinations were tried in King (1998); the perplexities of these dialogue models are given in table 4.6. The model that reduces the perplexity the most uses the speaker identities and the other person's previous move, as illustrated in figure 4.5. This seems intuitive as one is more likely to be responding to the other person's move, rather than the previous move which may have been one's own.

- ***Backing off***

Again not all possible sequences of the predictors in model III occur in the training set, therefore backing off techniques were tried. Using a floor value for low frequency combinations did not result in a reduction in perplexity. One can infer

Model	Predictors	Test Set Perplexity
simple trigram	m_{i-1},m_{i-2}	6.1
simple 4-gram	m_{i-1},m_{i-2},m_{i-3}	6.8
4-gram model I	m_{i-1}, s_i, s_{i-1}	5.5
4-gram model II	m_{i-2},m_{i-1}, s_i	6.2
4-gram model III	m_{i-j}, s_{i-1}, s_i	**5.2**
4-gram model IV	m_{i-2},m_{i-1}, s_{i-1}, s_i	5.8

Table 4.6: Perplexity results for the different dialogue models

	Information source	Move type accuracy (%)
A	Baseline	24
B	DM only	37
C	Recogniser output and LM	40
D	Recogniser output and LM and DM	**57**

Table 4.7: Move detection results using various information sources in the overhearer scenario

from this that the N-grams that do not occur in the training set also do not occur in the test set. Therefore, using other more complicated backing off techniques, such as those described in section 4.2.2, is unlikely to improve the predictability of the language models. Furthermore, backing off of mixed predictor dialogue models is more complicated. In the example given for trigram language models, the left most word is dropped. Choosing which predictor to drop is more complicated for mixed predictor models and is not pursued in this work.

4.6 Move Recognition Results

This section gives move recognition results using the language models and dialogue models described above. These experiments are conducted in the *overhearer scenario*, where the types of the previous moves are those predicted by the system as it goes through the dialogue, as described in section 3.2. Various move recognition results are given in table 4.7, each of which will be discussed in turn.

- ***Dialogue Models***

Using the 4-gram model alone yields a move recognition result of 37% (experiment B in table 4.7). Using a unigram gives a 24% recognition, which is equivalent to the baseline since the system chooses *acknowledge* for each utterance as this is the most frequent type.

- ***Recognition Output and Language Models***

As discussed above, the recogniser is run in conjunction with each of the move specific language models in order to determine the most likely move type of a given utterance. This method of automatic move classification gets 40% of the move types correct (experiment C table 4.7). This figure is lower than those given in the Verbmobil result of (67% for German and 72% for English). Those results, however, are derived using the transcribed word sequences. The results are also not directly comparable as the number of moves and the baseline differ for the different corpora.

- ***Recognition Output, Language Models and Dialogue Models***

The likelihoods from the language models are combined with probabilities from the dialogue model in a viterbi search. This is formalised in equation 4.16 for finding the most likely move sequence $M\star$. The dialogue component calculates the a priori probability of a move given the speaker sequence S. The language model is used to calculate the likelihood of each move given the cepstral observation sequence C. For a more detailed description of the system see chapter 3.

$$M^{\star} = \max_{M} \underbrace{P(M|S)}_{\substack{\text{dialogue} \\ \text{model}}} \cdot \underbrace{P(C|M)}_{\substack{\text{speech} \\ \text{recogniser}}} \tag{4.16}$$

This method yields a 57% recognition accuracy result (experiment D in table 4.7).

4.6.1 Conclusion

In this chapter, it has been shown that the system can perform automatic move recognition with reasonable accuracy using dialogue models and language models in conjunction with the recognition output. Once move recognition has been performed, the system chooses which sub-language model to use during word recognition. These word recognition results are given in chapter 7.

Chapter 5

Automatically Extracting Intonation Features

5.1 Introduction

One of the main goals of this thesis is to be able to automatically detect the type of dialogue act represented by an utterance. Intonation can be indicative of the function of an utterance in the discourse. For example, an utterance with declarative word order such as "You have a totem pole" can be a statement with a falling boundary or a question with a rising boundary. This illustrates that just modelling the wording as discussed in the previous chapter would lead to errors that only a model of intonation would be able to detect.

Chapter 6 approaches the subject of how one can predict move types using statistical models of intonation. This chapter compares using three different statistical models: classification trees (CART), hidden Markov models (HMM) and artificial neural nets (ANN). In order to train these models, a set of intonation features that potentially relate the contour to discourse function have to be extracted. There are two approaches to finding these features. Firstly, one appeals to the literature to find a linguistically motivated set of features. HMMs are trained solely on a set of these features known as the *tilt parameters*.

Secondly, a set of more global features, based on the study by Shriberg *et al.* (1998) are extracted e.g. mean F0 and energy. These features, listed in section 5.4 are used along with the theoretically motivated features in the CART and ANN models. By examining the CART tree, one can see which of these features are used to discriminate utterance types. Section 6.4.4 will show that the tree uses both the theory specific features, such as the tilt value or shape of the final accent and the more general features, such as utterance duration and F0 mean.

5.2 Intonation Analysis

In order to be able to distinguish different types of moves, one has to identify the distinguishing features in the intonation contours of utterances of the same type. There are two areas of research that are discussed in the literature review. Firstly, a standard method of analysing contours is needed in order to identify the similarities and differences of contours associated with a specific move type. Secondly, an algorithm is sought after that can be used to perform this analysis of the intonation contour automatically. This way, the features can be used to train the statistical intonation models that perform automatic move classification. Details of these intonation models are given in the following chapter.

5.2.1 IPO

The model of Dutch intonation developed at Institute for Perception Research (known as IP0) defines the contour as a sequence of intonation events (t'Hart & Cohen, 1973). These events are categorised in terms of their pitch *movement*, (rise or falls). They classify events into two categories: *prominence lending* and *non-prominence lending*. Prominence lending pitch movements are associated with a prominent syllable, for example a rise or a fall early on the stressed syllable. The sequences of rise and falls form phonologically distinct contours. That is to

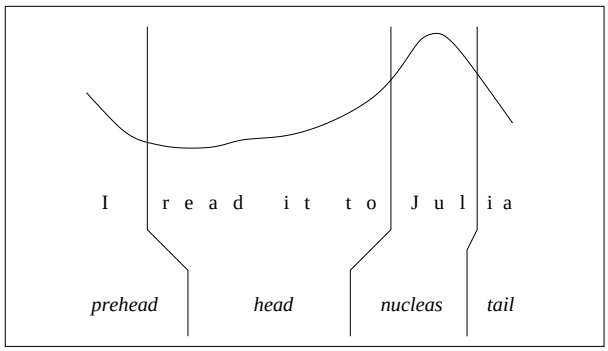


Figure 5.1: Division of the intonation contour by the British School. Source: Ladd (1990)

say, the same sequence of phonological events may be phonetically realised in a number of ways depending on certain factors, for example utterance length.

Non-prominence lending movements are important to the current study as they can be indicative of utterance types. They include movements at the boundary of utterances and rise or fall movements that span more than one syllable. Boundary tones are thought of as phonologically distinct, as different boundary tones turn utterances into different types.

5.2.2 British School of Intonation Description

The British School of Intonation Description e.g. Palmer (1922), O'Connor and Arnold (1973) and Crystal (1969), also describe the intonation contour as a sequence of movements. They add a further layer of analysis by dividing the contour into different parts: pre-head, head, nucleus and tail, as illustrated in figure 5.1, taken from Ladd (1996). The nucleus is the only tone group that is not optional. Although originally the IPO model did not have such a delimitation of the contour, they have recently developed a similar structure analysis of prefix, root and suffix (t'Hart *et al.*, 1990).

5.2.3 Autosegmental-metrical Theory of Intonation

The Autosegmental-metrical theory (AM) is similar to the IPO and the British School in that the intonation structure is defined as a linear sequence of intonation events. However, the AM theory defines pitch movement in terms of tones or pitch targets (High or Low) as opposed to rises or falls.

There are three types of intonation events: pitch accents, phrase tones and boundary tones. Pitch accents are either H for high level tone or L for low level. If a tone is associated with a stressed syllable it is accompanied by a star (e.g. H*). A tone may be accompanied by a leading or a trailing tone. For example, a rising accent in IPO terminology is a L*+H or L+H* in AM terminology. Boundary tones are transcribed with a percent sign (e.g. L%) and mark the end of a large intonation phrase. Phrase accents or tones are unstarred Hs or Ls that are unattached and come between the last pitch accent and the boundary tone. Different intonation events in the same category are phonologically distinct.

This taxonomy can be given in terms of the finite state (FSN) grammar illustrated in figure 5.2 taken from Pierrehumbert (1980). Each node of the FSN corresponds to the different types of events. The FSN generates all the possible intonation contours that occur in English. One can see that, unlike the British School's analysis, Pierrehumbert gives no theoretical validity to the status of nuclear accent. Her theory is based on defining the contour in terms of a sequence of events generated by the FSN. She argues against any sort of global contour shapes such as the British School's.

One argument against including the nuclear status is that there is no phonetic difference between these accents and prenuclear accents, as shown by Silverman & Pierrehumbert (1990). Ladd (1996) argues, however, that although there may be no phonetic difference, this is not an argument against the idea that nuclear accents play an important role in the phonological structure of contours. Ladd

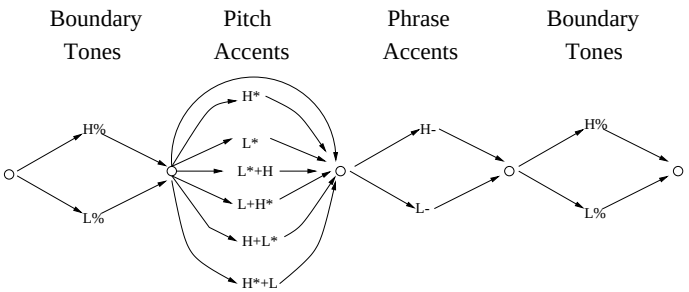


Figure 5.2: FSN for all possible event combinations. Source: Pierrehumbert (1980)

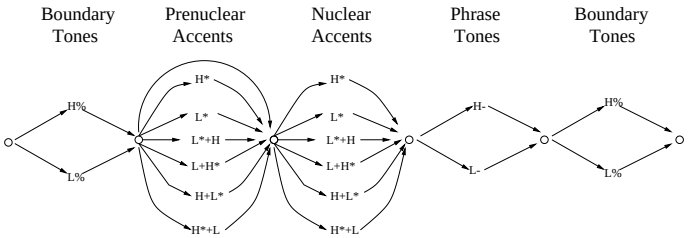


Figure 5.3: FSN for all possible event combinations. Source: Ladd (1996)

.

amends Pierrehumbert's FSN by adding an obligatory node for the nuclear accent illustrated in figure 5.3. This preserves Pierrehumbert's notion that the string consists of a sequence of events generated by the FSN, while still giving separate status to the nuclear accent.

5.2.4 ToBI

ToBI (Silverman *et al.*, 1992) is a version of Pierrehumbert's taxonomy that is a proposed standard for intonation labelling of electronic English databases[1]. The ToBI labelling system consists of a number of tiers, including orthographic transcription. One of the tiers marks the event type (To) in terms of Hs and Ls and one gives a measure of phrase breaks between each word (BI).

[1]There are language and dialect variants such as GToBI for German (Grice *et al.*, 1996) and GlaToBI for Glaswegian (Mayo *et al.*, 1997).

5.2.5 Verbmobil

The set of prosodic labels used in the Verbmobil project, described in section 1.4, consists of four types of boundary tones which correspond to the break indices of ToBI. They define three types of accents based on the starred accents in ToBi. For automatic classification of accents, they use a vector of features which are related to the output of the speech recogniser, e.g. speaking-rate, duration of filled and silent pauses and duration of segments. However, characterising intonation features in terms of the output of the recogniser leads to inaccuracies; see Bucklow *et al.* (1999) for more details.

5.2.6 Fujisaki

All of the systems described above are *phonological*. In other words, they consist of a sequence of phonological events which can have various *phonetic* realisations depending on the context. An alternative model has been proposed by Fujisaki (1982) which is non-phonological and quantitative. This model defines the F0 contour as a complex function consisting of two component F0 functions, namely a phrase component and an accent component. The accent component defines the local perturbations corresponding to intonation events. The phrase component incorporates utterance level effects such as declination.

Ladd (1996) states that this type of *overlay model* can be useful for modelling local perturbations and overall pitch range changes due to, for example, changes in emotion. Ladd highlights one problem concerning the phrase component. This identifies the phrase boundaries that result in the ideal model of the F0 but the boundaries do not always correspond to the linguistic structure of the utterance. This results in a model of the physical aspects of F0 but which makes no sense linguistically.

5.2.7 The Tilt Model

The *tilt* model is also a non-phonological quantitative model and was developed by Taylor (2000; 1998), to provide an intonation model for speech synthesis and recognition. Taylor attempts to define a model that is reasonably constrained to avoid overgeneration and yet still be able to cover a wide variety of observed phenomena.

Instead of using such discrete labels as the ToBI system to classify intonation events, Taylor uses continuous variables known as the *tilt parameters*. These parameters are:

- *tilt*: shape of the event
- *F0 amplitude*, a measure of the F0 excursion of the event
- *start F0*: the F0 value at the start of the event
- *duration*: length of the event in time
- *peak position*: position of the event peak

The above continuous variables are used instead of discrete categories to classify events for a number of reasons. Firstly, classifying accents into ToBI labels is a difficult task, even for human labellers. In a study based on the ToBI labelling scheme (Pitrelli *et al.*, 1994), labellers agreed on pitch accent presence or absence 80% of the time, while agreement on the category of the accent was just 64%; this figure was only achieved by first collapsing some of the main categories (e.g. H* with L+H*). Similarly, the system described in Taylor *et al.* (1997) finds the task of locating pitch accents much easier than classifying them.

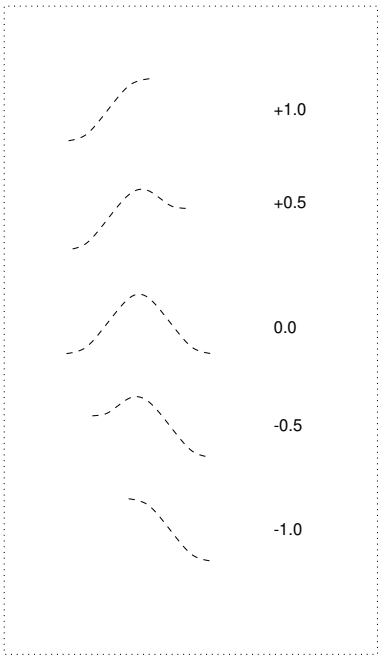


Figure 5.4: Values for tilt for various shaped intonation events

- ***Interpreting the Tilt Parameters***

The tilt value is a figure between -1 and 1 and describes the shape of the contour. Examples of varying tilt values are given in figure 5.4 taken from Taylor (2000). The other four tilt parameters (start F0, F0 amplitude, duration and peak position) are shown in figure 5.5.

F0 amplitude is given in Hertz and is a phonetic measure of prominence. However, this level of prominence is dependent on the position in the utterance. Liberman and Pierrehumbert (1984) show that pitch range decreases nearer the end of the utterance. The same measure of amplitude corresponds to less prominence if it is near the start than if it is near the end of the utterance. Therefore, further processing is necessary to obtain a prominence value that has phonological significance.

The start F0 parameter is necessary for analysis and resynthesis. Duration is given in seconds. The length of an event can be affected by a number of factors

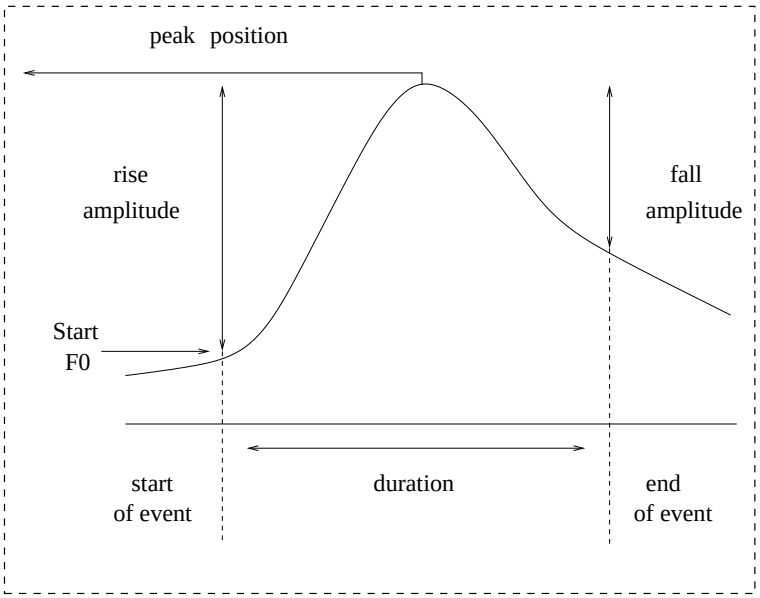


Figure 5.5: Rise fall accent showing the tilt parameters, excluding tilt

including the F0 contour, the segmental string and speaking rate.

- ***Peak Position***

Taylor takes peak position as the distance in time from the start of the utterance to the peak, as illustrated in figure 5.5. This distance figure is necessary for synthesis but is not an intonationally significant feature. Taylor proposes an alternative measure called the *syllabic position parameter*, which is the distance from the peak to the start of the vowel in stressed syllable. This would provide a parameter that is similar to the other tilt parameters in that it is locally orientated. More importantly, it would capture a distinctive feature of accents associated with accent *alignment*. Alignment is the phonetic realisation of the accent with relation to the stressed syllable.

At the time of the experiments described in this thesis only the absolute peak position was derivable from the wave form. Subsequent experiments described in chapter 9 look at ways of deriving the more linguistically useful measure of syllabic position and examine its relation to the other tilt parameters.

5.3 Automatic Analysis of the Intonation Contour

The studies described above provide a method of analysing the contour in terms of intonation events which are classified using discrete labels or continuous variables. These methods will not be useful in the system described in chapter 3, unless the feature extraction is fully automated.

There have been a number of studies that attempt automatic intonation analysis. These vary depending on whether they are attempting to identify and classify pitch accents or locate stressed syllables or segments e.g. Hieronymus (1989). Approaches also vary depending in their applications, whether they are used for automatic labelling of data or in an automatic recognition system. The first of these applications usually assumes segmental transcription, For example Wightman & Ostendorf (1994) train a decision tree using syllable level features for accent classification. Methods of automatically identifying and classifying intonation events based on the waveform alone are of interest here.

Ross & Ostendorf (1995) attempt accent location and classification in terms of ToBI labels for recognition. They use F0, energy and syllable duration to train a statistical model. Their result for accent location accuracy is 85%. Overall classification accuracy is 65%. Distinguishing rising and falling boundary tones is reasonably accurate (85%). The disadvantage of this system is that it assumes fixed syllable boundaries which it uses with the recogniser to align the segments. This syllable information could be obtained by various automatic techniques, all of which are error prone.

In a module in the Verbmobil system, Bucklow *et al.* (1999) use a multi-layer perceptron (MLP) to categorise intonation events into accents and boundary tones after they have been identified. The MLP is trained on a set of prosodic features for each syllable. A description of these features is given in section 5.2.5. The

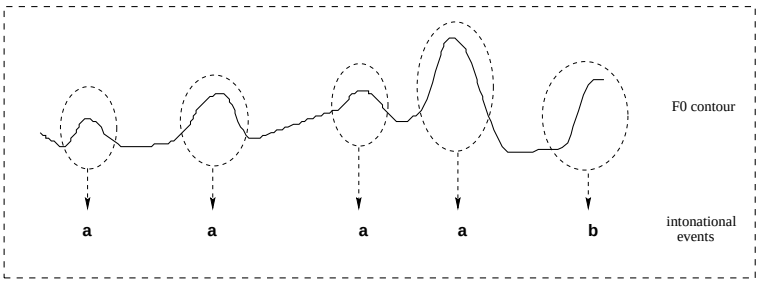


Figure 5.6: Intonation contour with labelled accents and boundaries corresponding to the circled pitch excursions

MLP distinguishes accents from boundaries with an accuracy of approximately 80%.

Taylor (2000) provides a method of intonation analysis that requires no pre-processing and is totally automated. His algorithm is divided into two main processes. The first is the automatic placement of accents and boundary tones. Accents are marked with "a"; "m" marks a minor accent. Boundaries "b" can either be falling "fb" or rising "rb". There is an additional category "ab" for use when accents and boundaries occur too close to be separated. The second process automatically defines these events in terms of the five *tilt parameters* discussed above.

5.3.1 Automatic Event Detection

Intonation event detection is performed using a continuous density HMM system. Each utterance is represented acoustically by F0 and energy, and their first and second derivatives. Separate context independent models are trained for accent and boundary type events. The system is trained on set B of the DCIEM corpus (see the data section 3.2), which is hand-labelled for these intonation events. A schematic representation of intonation labelling is given in figure 5.6, taken from Taylor (2000) and modified.

Performance is assessed by measuring how well the hand-labelled test set

matches the output of the recogniser. For an automatically labelled event to count as correct, it must overlap a hand-labelled event by at least 50%. Using this metric, the performance of the recogniser is 86.5% correct with 54.3% accuracy. An equivalent speaker dependent system trained on part of the data gave 87% correct and 63% accuracy. Taylor is currently developing speaker normalisation techniques which may improve the accuracy of the speaker independent model.

5.3.2 Aligning the Rise and Fall Parts of Events

In order to calculate the tilt parameters, the start and end points of the accent must be identified. To do this, the approximate location of an event given by the HMMs and the F0 contour are used. Firstly, the F0 between the event boundaries is smoothed and the unvoiced regions interpolated (see figure 5.7a). This helps to factor out spurious perturbations and speaker affects. A peak-picking algorithm is used to determine the rise and fall parts of the event. This is known as RFC analysis, illustrated in figure 5.7b (Taylor, 1995).

For single rise and falls, a search region of 20% of the event length is set before and after the potential start and end points of the accent. The curve is resynthesised for each of these potential boundary frames and the Euclidean distance[2] from the original curve is calculated. For rise-fall contours the rise section is aligned first. With a fixed peak position and start point established, the fall part of the contour is then aligned.

5.3.3 Automatic Tilt Analysis

The *tilt* parameters describe the type of event and are calculated in two stages. Firstly, the F0 amplitude and duration values in figure 5.7c are calculated by equations 5.1 and 5.2 respectively.

[2]This distance is a measure of good fit.

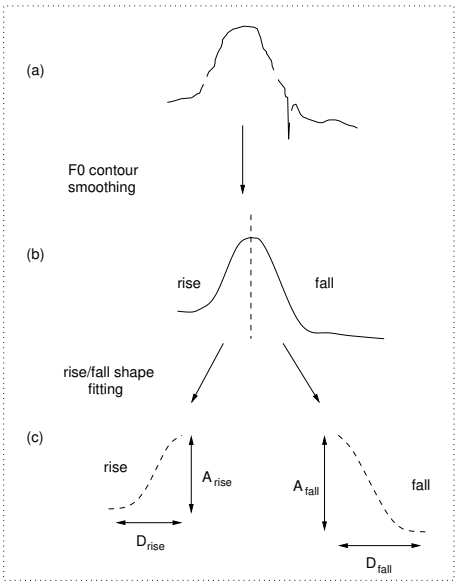


Figure 5.7: a) shows F0 smoothing; b) rise/fall fitting using the RFC model; c) amplitude and duration calculations used in the tilt analysis. Source: Taylor *et al.* (1998).

$$tilt_{amp} = \frac{|A_{rise}| - |A_{fall}|}{|A_{rise}| + |A_{fall}|} \tag{5.1}$$

$$tilt_{dur} = \frac{D_{rise} - D_{fall}}{D_{rise} + D_{fall}} \tag{5.2}$$

These equations can be combined, as they are highly correlated. The equation for tilt is given below.

$$tilt = \frac{A|_{rise}| - |A_{fall}|}{2(|A_{rise}| + |A_{fall}|)} + \frac{D_{rise} - D_{fall}}{2(D_{rise} + D_{fall})} \tag{5.3}$$

Finally, the F0 amplitude and event duration are calculated in equations 5.4 and 5.5 respectively.

$$A_{event} = |A_{rise}| + |A_{fall}| \tag{5.4}$$

$$D_{event} = D_{rise} + D_{fall} \quad (5.5)$$

5.4 Other features

So far this chapter has been devoted to looking at various ways of extracting intonation features automatically from the waveform. In particular, a method was presented for automatically identifying accents and extracting their distinctive features. A separate set of more general features were also extracted from the waveform. This set of features is based on features used in a similar study described in Shriberg *et al.* (1998).

These features fall into the three main categories which are described in detail in the following sections:

- F0 features (e.g. max F0, F0 mean and standard deviation, least squares regression line)
- Energy features (e.g. energy mean and standard deviation)
- Duration features (e.g. number of frames in utterance, number of frames of F0)

5.4.1 F0 features

The list of features involving F0 is given in table 5.1. The first set of features captures general characteristics of the utterance, for example the standard deviation of the F0 represents pitch range. As the final part of the intonation contour is often indicative of utterance type, calculations are made for the last and penultimate 200ms[3] of the utterance as well as for the whole utterance (e.g. end_F0, pen_F0).

[3]This figure is chosen based on the study described in Shriberg *et al.* (1998).

Feature Name	Description
max_F0	utterance max F0
utt_F0_mean	utterance mean F0
utt_F0_sd	utterance standard deviation F0
end_F0_mean	end region F0 mean
pen_F0_mean	penultimate region F0 mean
norm_end_F0_mean	end region F0 mean normalised using the utterance mean and sd
norm_pen_F0_mean	pen. region F0 mean normalised using the utterance mean and sd
abs_f0_diff	difference between mean F0 of end and penultimate region
rel_f0_diff	ratio mean F0 of end and penultimate region
norm_f0_excursion	ratio of F0 sd of end region over utterance
utt_a, utt_b	least-squares all-points regression line over utterance
end_a, end_b	least-squares all-points regression line over end region
pen_a, pen_b	least-squares all-points regression line over pen. region
type_boundary	Type of final boundary (falling, rising)
num_acc	number of accents
num_bound	number of boundaries
num_acc_bound	number of accent and boundaries
total_num_abs	total number of accents

Table 5.1: F0 feature list

The features in the next set are calculated by comparing feature values for the two end regions and the whole utterance, e.g. ratio of mean F0 in the end and penultimate regions (rel_f0_diff). In addition to these features, the least-squares regression line of the F0 contour is calculated for the last and penultimate 200ms and for the whole utterance. This would capture intonation features such as declination over the whole utterance, and boundary type over the final part of the contour. This is illustrated in figure 5.8.

For the least squares regression line y is $y = ax + b$, the values a and b are calculated by solving the following simultaneous equations

$$\sum y = a \sum x + nb \tag{5.6}$$

$$\sum xy = a \sum x^2 + b \sum x \tag{5.7}$$

x refers to time frames in seconds and y corresponds to the F0 value at that time.

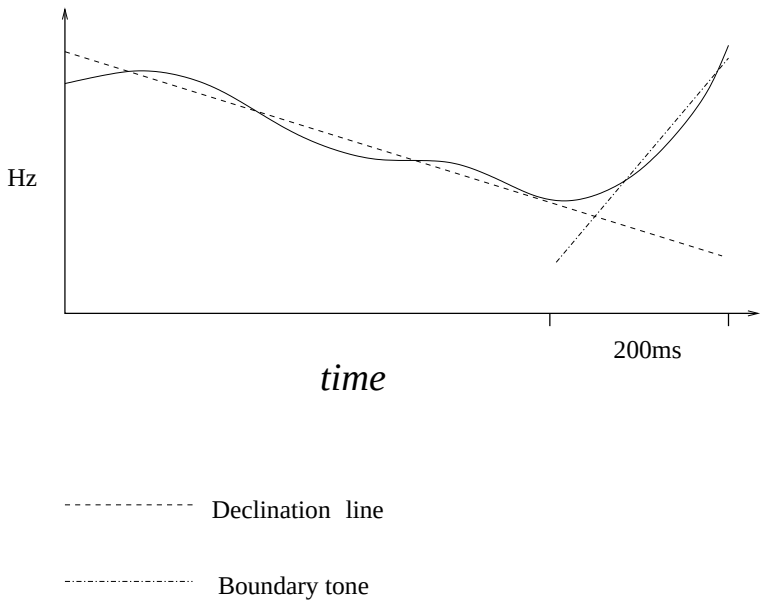


Figure 5.8: Intonation contour with least square regression lines capturing the line of declination and the boundary tone

Type_boundary is a binary value given for the type of boundary (1 for rising and 0 for falling), depending on the tilt value of the final accent. The number of accents (a), boundary (b) and joint accent boundary tones (ab) were counted (num_acc, num_bound, num_acc_bound, total_num_abs).

5.4.2 Energy features

A general set of features is calculated for the root mean squared (RMS) energy values. These are given in table 5.2.

5.4.3 Duration Features

There are three duration features listed in table 5.3. Utterance duration is the number of frames of the utterance including utterance initial and final silences and voiceless segments. F0_length is taken from the start to the end of voicing and includes voiceless sections. Regr_num_frames is the number of frames containing voicing, used to calculate the F0 regression line for the whole utterance.

Feature Name	Description
utt_nrg_mean	mean RMS energy in utterance
utt_nrg_sd	standard deviation RMS energy in utterance
end_nrg_mean	mean RMS energy in end region
pen_nrg_mean	mean RMS energy in pen. region
norm_end_nrg_mean	mean RMS energy in end region normalised over utterance
norm_pen_nrg_mean	mean RMS energy in pen. region normalised over utterance
abs_nrg_diff	difference between mean RMS energy at end and pen. regions
norm_nrg_diff	difference between norm_end_nrg_mean and norm_pen_nrg_mean
rel_nrg_diff	ratio of end_nrg_mean and pen_nrg_mean

Table 5.2: Energy feature list

Feature Name	Description
utt_duration	number of frames of whole utterance
f0_length	duration of F0 contour in seconds, including voiceless frames
regr_num_frames	number of frames of F0 contour, excluding voiceless frames

Table 5.3: Duration feature list

5.5 Summary

The aim of this chapter is to give an outline of certain methods of evaluating distinctive intonation features. An account of the *tilt* model was given. This is an alternative approach to the classic discrete labelling systems, such as ToBI. This system is totally automatic and provides a set of continuous variables that can be used in a statistical model, such as the hidden Markov models described in the following chapter. The tilt parameters of the final three accents in a phrase are combined with the 33 general features given in the last section of this chapter. All these features are used to train classification and regression trees and artificial neural nets, also described in the following chapter. These models determine the most discriminating features for classifying utterances into the different move types. Examining the features used by the CART model shows that the process of extracting tilt features is justified and that general features such as utterance length and maximum F0 are also used to determine move type.

Chapter 6

Intonation Models

6.1 Introduction

There are two questions approached in this chapter, firstly whether intonation is indicative of move type and secondly, if it is, whether characteristic intonation patterns can be effectively modelled using statistical techniques.

The previous chapter looked at ways of classifying certain features of the intonation contour. Specifically, the contour is represented as a sequence of intonation events which are categorised in terms of pitch movement, pitch targets or a set of five continuous variables, depending on the theory adopted. The different sequences of events are known as *tunes*. The literature review in section 6.2 examines previous attempts at linking these tunes to functions in the discourse, in terms of dialogue acts, speech acts, attitude, emotion, etc..

In section 6.3, a novel method is presented that models intonation at an utterance level by using statistical methods. These statistical intonation models fit into the framework described in chapter 3 for automatic move detection. Section 6.7 gives move detection results using the intonation models in conjunction with the dialogue models described in chapter 4. Results for move detection and word error rate for the system as a whole are given in chapter 7.

6.2 Mapping Intonation to Discourse Function

This section gives an overview of previous attempts at mapping intonation contours or *tunes* to discourse functions. The approaches described below fall into two categories: a bottom-up and a top-down approach. The bottom-up approach takes the intonation contour and attempts to assign discourse meaning. The alternative approach looks at the different functions in discourse and examines how these are realised in terms of intonation features.

6.2.1 The Top-down Approach

Sag and Liberman (1975) identify certain intonation contours in an attempt to separate indirect speech acts from direct ones, such as contours that distinguish questions from suggestions and other utterance types. They establish that certain contours force a literal interpretation, such as using their "tilde" contour with wh-questions. However, they find that some contours such as their surprise redundancy contour can be used to express direct and indirect acts, depending on the context in which it is used.

Kowtko (1996) examines whether discourse function correlates with the type of intonation pattern. She finds that the intonation of a move can vary, depending on a number of factors. For example, the intonation of *acknowledge* type moves varies depending on which game it is in. Intonation of moves also vary depending on whether the data are read or spontaneous.

6.2.2 The Bottom-up Approach

The bottom-up approach focuses on prosodic characteristics, whereby intonation categories are identified and associated with a function in the discourse. This is the basic approach of O'Connor & Arnold (1973), who describe ten canonical tunes with names such as *low bounce* and *high drop*. Each tune has various functions and

attitudes. For example the *high drop* tone group is used for statements that have an indication of "warmth". However the *high drop* is also used in wh-questions, yes-no questions and commands. Examples of these are given below, taken from O'Connor & Arnold (1973) (\ indicates a high drop on the following word).

Statement:	(What time is it?)	**It's half past \twelve. I didn't realise how \late it was.**
Wh-question:		**What's the \time?**
(Yes-no question	(John says he's got an alibi.)	**Can he \prove it?**
Command	(What shall I do with this rubbish?)	**\Burn it**

Similarly Liberman (1975) defines tunes in terms of speaker attitude such as the *surprise/redundancy* tune. When this tune is used with a statement, it conveys an expression of surprise or that the propositional content is, or should be obvious. Gussenhoven (1984) and Crystal (1972) attempt to associate meaning, emotion and intention to intonation contours.

The studies mentioned above take a holistic approach to tune classification. Pierrehumbert & Hirschberg (1990) attempt a "compositional theory of tune interpretation". The authors argue that the different features of a tune (pitch accents, boundary tones, phrase accents) convey certain aspects of meaning. Tunes that have certain parts in common share some of the same meaning. For example, L*+H L H%, H* L H% and L+H* L H% all share a low phrase tone and a high boundary tone that indicate a following utterance will complete the speaker's intended meaning.

The main problem with this bottom-up approach is that intonation tunes do not map directly onto the utterance types. For example, the H* L L% is frequently used with declaratives but also can be used with wh-questions.

6.3 Statistical Models of Intonation

Both of the approaches discussed above are problematic due to the many-to-many mapping of the intonation contours and discourse functions of utterances. I propose a solution to this problem by using the top-down approach and developing statistical models of intonation that can model the variation associated with the different types of move. The statistical models take into account the mean and variation of the prosodic features of a move type. A long as the probability distributions for the features are different for each of the move types, the models will be able to determine the type of a given utterance.

Intonation plays a crucial role in the automatic classification of moves. For example, if one has a phrase of declarative syntax uttered with a rising boundary tone (e.g. "you have a totem pole?"), the speech recogniser module of the system would classify this as a statement. The dialogue model may classify it as a question if it is followed by a reply. The intonation model would typically give a high likelihood of it being a question, thus outweighing the speech recogniser module and classifying the utterance as an interrogative.

The statistical models fit within the framework described in chapter 3, as they produce the likelihood of the intonation given the different move types, i.e. $P(I|M)$ in the follow equation:

$$M^\star = \max_M \underbrace{P(M|S)}_{\substack{\text{dialogue} \\ \text{model}}} \cdot \underbrace{P(C|M)}_{\substack{\text{speech} \\ \text{recogniser}}} \cdot \underbrace{P(I|M)}_{\substack{\textbf{intonation} \\ \textbf{model}}} \tag{6.1}$$

where I represents the intonation features and M the move type (see section 3.4.1 for more details).

Three stochastic methods are examined for modelling the intonation contour. These are:

- Classification and Regression Trees (CART)
- Artificial Neural Nets (ANN)
- Hidden Markov Models (HMMs).

Each of these models will be discussed in turn and their effectiveness as intonation models compared.

6.4 Classification and Regression Trees

This section reports work using classification and regression trees (Breiman *et al.*, 1994) to automatically predict move types. These are binary decision trees trained on many intonation features. Much of this work is inspired by the use of classification and regression trees trained for utterance type detection on a large database of telephone conversations, reported in Shriberg *et al.* (1998) and Jurafsky *et al.* (Jurafsky *et al.*, 1997), who use similar features to train a classification tree to differentiate utterance types with reasonable success.

6.4.1 Previous use of CART Trees in Synthesis and Recognition of Intonation

Regression trees have been used to predict values for prosodic features for the resynthesis of F0 contours. The experiments described in Dusterhoff (2000 (forthcoming)) involve developing a set of trees for each type of intonation event (e.g. "a", "b") trained on 45 prosodic features available during text-to-speech synthesis. These decision trees are used for the prediction of tilt parameters for a given sequence of events. A similar preceding study (Black & Hunt, 1996) involves training regression trees to predict the F0 contour for every syllable.

From a recognition perspective, CART trees have been used for identifying disfluencies (Shriberg *et al.*, 1997), repairs (Hirschberg & Nakatani, 1993), and discourse cue phrases (Grosz & Hirschberg, 1992). Shriberg *et al.* (1997) use CART trees trained on prosodic features to determine for each word boundary whether there is a repair of various kinds: filled pauses, repetitions, repairs or false starts. The features include durational features, distance from last pause, F0 and energy features. They combine this prosodic model with a language model trained to find boundaries as word tokens (Stolcke & Shriberg, 1996). The trees trained for classifying the various types of repairs are reasonably successful, with the following accuracy rates[1]: filled pauses, 89.7%; repetitions, 77.5%; repairs, 75.5% and false starts, 74%.

Hirschberg & Nakatani (1993) use a mixture of prosodic features and word level features to train a CART tree to find interruption sites of self-repair. Examples of these features include: pause distance between words, normalised energy of word, F0 of word in relation to the following and previous word, filled pauses, part of speech information. This CART tree identifies 86% of the interruption sites correctly with an accuracy of 77%.

Grosz & Hirshberg (1992) use CART trees trained on prosodic features to identify discourse features in read speech based Grosz and Sidner's model of discourse structure (Grosz & Sidner, 1986).

These studies show that decision trees can be useful in modelling prosodic features, for a number of purposes. I will show that they can be used to model intonation features at an utterance level for the classification of utterances into the 12 move types. CART trees are particularly useful as one can examine the features used in the decision tree to obtain an insight into the predictive nature of the data.

[1]See section 3.4.4 for a definition of accuracy.

6.4.2 Training CART Trees

There are three main rules involved in training the tree:

1. A rule for selecting the best split in a tree
2. A rule to stop tree growth
3. A rule for assigning every terminal node to a class

Each of these rules shall be discussed in turn.

1. The first rule decides how the data will be split to form groups that are the most similar. Splits are in the form of questions. During training, one needs a method of checking how effective the trees are at final classification. This error function, or misclassification rate is defined as the probability that a pattern will be allocated the wrong class. There are several methods of estimating this error rate. One method involves extracting a portion of the training set, known as the "held out" set. This set is randomly chosen from the training set in order to maximise independence. Testing the progress of the training of the tree using the held out set does improve its performance.
2. A CART tree does not have a fixed size. A method is therefore needed for stopping the growth of the tree at some point or the final tree will have one data point at each node. The method described in Breiman *et al.* (1994) involves growing the tree until terminal nodes are very small and then pruning the tree upwards getting a sequence of subtrees. A held-out test set is used to pick out the subtrees with the lowest misclassification rate.
3. When the tree has stopped growing the categories in the leaves are examined and a leaf is assigned a category or a figure relative to the majority of members.

6.4.3 Experiments Using the DCIEM Data

This section describes how a classification decision tree is trained to classify utterances into the 12 different types. Forty-five features are automatically extracted from the speech signal for each utterance in the training set A. These include the 33 general features given in section 5.4, plus the tilt parameters for the last three accents (if present) in the contour. Peak position of these accents was not included in these features as it was thought that it has no linguistic significance, see chapter 9 for further discussion. All of these features are normalised to fall within -1 and 1.

The features are used to train one classification tree. The output of this tree is a set of likelihoods for the 12 move types. These likelihoods are actually the posterior probability of the moves given the intonation features $P(M|I)$. In other words, the tree takes into account the distribution of the moves in the training set. In order to use a CART tree in the system described by equation 6.1, one needs to compute the likelihood of the intonation given the different move types $P(I|M)$. There are two methods of doing this. The first is to divide the output of the tree by the prior probability, $P(M)$:

$$P(I|M) = \frac{P(M|I)\dot{P}(I)}{P(M)} \tag{6.2}$$

An alternative method is to train the tree on data containing equal numbers of moves. A certain number of examples for each move were randomly selected from the training set. The move type with the lowest frequency is *clarify* with 93 occurrences. Therefore, 93 examples of each move type were used, resulting in a training set containing 1113 different utterances. In order to produce a larger training set, duplicate sets of intonation features were used for moves with lower frequency. This "equal data" method produced slightly better results; therefore results quoted henceforth are obtained using trees trained on data with 200 entries for each move type.

6.4.4 Tree Interpretation

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, the features are grouped into 3 general categories of duration, F0 and energy. For a detailed description of the features that make up these categories, see section 5.4. Table 6.1 gives the *feature usage frequency* for these groups of features. This measure is the number of examples in the training set classified using each feature. 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 with respect to the total number of training examples to give a percentage.

Different move types by their nature vary in length; therefore it is not surprising that duration is highly discriminatory in classifying utterance types. For example, *ready, acknowledge, reply-y, reply-n* and *align* are distinguished from the other moves by the top node which queries a duration feature.

The top 10 features are given in table 6.2. The duration feature used most is regr_num_frames, which 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 may be a fairer measure of actual speech duration than the other duration features, as it excludes pauses and silences.

The F0 feature that is queried the most is the F0 mean in the end region of the contour (end_f0_mean). The next top two F0 features are the maximum F0 and the point where the least squares regression line for the last part of the utterance crosses the y-axis (end_b). The use of these features indicates that the F0 contour near the end of the utterance contains important linguistic information for the

Feature Type	Usage (%)
F0	46
Duration	36
RMS Energy	1

Table 6.1: Discriminatory features and type usage in move classification trained using equal number of moves

Feature	Usage (%)
regr_num_frames	31.5
end_f0_mean	16
max_f0	8.5
utt_duration	7.9
end_b	7.3
abs_nrg_diff	6.7
auto_tilt	4.9
norm_auto3_amp	2.4
f0_length	2.4

Table 6.2: Top 10 features and feature type usage

distinction of move types.

There are two tilt parameters in the top 10 features queried. These are the tilt value of the final accent and the F0 amplitude of the third from the last accent. The tilt value of the final accent would be a clear indication of boundary type.

Figure 6.1 illustrates part of the classification tree. One can see how the tree splits the data, depending on the feature that is being queried[2]. The nodes represent the classification of the examples in the training set into the different move types.

The highest query in this part of the tree is whether the utterance has a high maximum F0. This may be indicative of a rising boundary tone or a large pitch range. All the utterances with this feature are classified as *query-yn*. If there is not a high maximum F0 but there is a high F0 mean in the last part of the utterance,

[2]The features are continuous variables, the binary terms such as *long/short*, are used in this figure for simplicity.

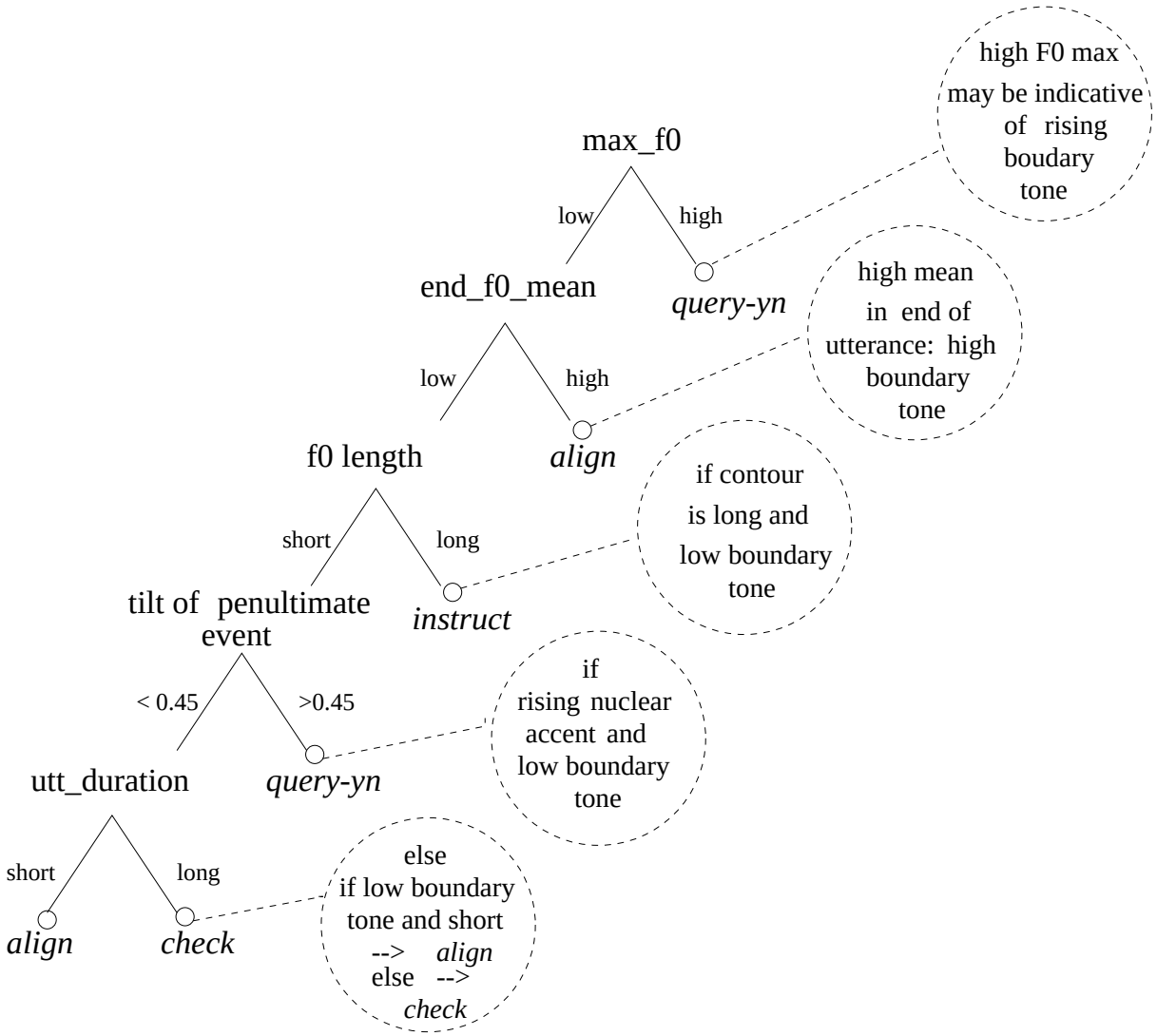


Figure 6.1: Part of the binary decision tree for classifying moves

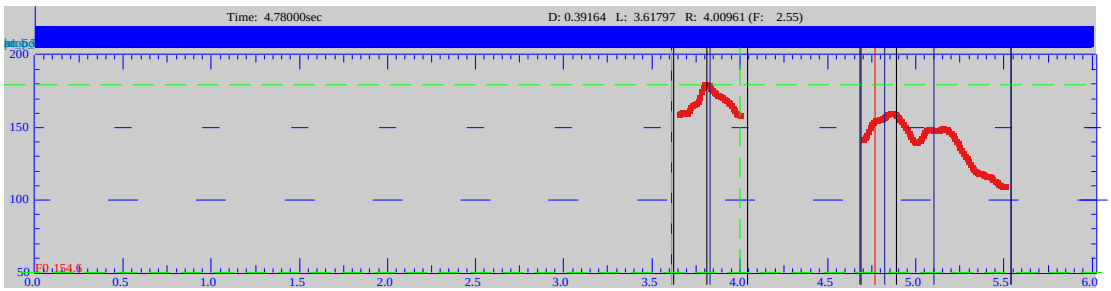


Figure 6.2: Example contour of a yes-no question move that has a falling boundary ("fb") and a high accent on the nuclear accent ("a") on "monastery". ("m" is a minor accent, see section 5.2.7)

then *align* and *query-yn* are given high likelihoods. These moves frequently have high boundary tones.

If an utterance does not have a particularly high boundary tone and the utterance is long, then the tree classifies it as an *instruct*. If it is not long and the penultimate[3] event is rising, indicated by a tilt value greater than 0.45, then the utterance is a *query-yn*. If one adopts the view of the British School of Intonation and takes the final non-boundary event to be the nuclear accent (see section 5.2.2), one can make the following generalisation: if a yes-no question has a falling boundary tone it is likely to have a high tone on the nuclear accent. If this was not the case the classification tree would not have used this feature to discriminate *query-yn* from the other types of moves, such as *check* and *align*. An example of a yes-no question with a low boundary tone and a high accent on the nuclear tone is given in figure 6.2. The F0 contour is accompanied by the word transcriptions and the hand-labelled accents where the delimitation lines indicate the mid point of the accent. For definitions of the labels see section 5.2.7.

[3]The penultimate accent is the last non-boundary event.

Finally the moves *check* and *align* with falling boundaries are discriminated by length. Recall from section 5.4 that "utt_duration" is the whole length of the utterance in terms of the number of frames, including utterance initial and final voiceless segments and silences. 'F0 length" is the duration in seconds of the F0 contour including only utterance internal silences.

This figure clearly shows that the classification tree is able to make sensible decisions by identifying features that are discriminatory in the classification of utterances into the 12 move types.

6.5 Neural Nets

Artificial neural nets (ANNs) are a machine learning algorithm used for classification given a vector of feature values. Like CART trees, they are particularly suited to the type of pattern recognition attempted in this thesis, namely using a vector of intonation features to determine the move type of an utterance.

The structure of artificial neural nets was motivated by the network of cells or neurons in the brain. However, they should really be viewed as simply a trainable statistical model. In this chapter, the structure of artificial nets will be discussed and the parameters set for this task specified. This section shows that ANNs are an effective method for classifying move type using suprasegmental features. The software used in these experiments is the Stuttgart Neural Network System (SNNS, 1997).

6.5.1 Appropriate Problems for ANNs

Early ANNs were in the form of single layer perceptrons (Rosenblatt, 1958). This type of neural net was unable to classify data that are not linearly separable. For example, they could model some of the rules of logic, such as AND, OR, but were unable to compute XOR (exclusive or). The development of a learning algorithm

for a multi-layer network, known as *backpropogation*, enabled ANNs to be used for a large variety of tasks.

Previous experiments have illustrated the effectiveness of networks that use the backpropagation learning algorithm to perform the classification of linguistic units. An example of one such study is NETtalk (Sejnowski & Rosenberg, 1986). This study involved training a multi-layer neural network for classifying graphemes into phonemes which are to be submitted to a speech synthesiser. The net has a classification rate of 81% correct.

ANNs have also been used for the task of recognising handwritten ZIP codes by LeCun *et al.* (1989). The input to the multi-layer net is a 16x16 digitised gray scale image and the output layer consists of 10 units, one for each numeral. They achieve an impressive error rate of 5% on the test set.

The above examples show that the task of classifying utterances into moves using a set of suprasegmental features as inputs is a plausible experiment.

6.5.2 The Perceptron

The net consists of many layers of *perceptrons*. A perceptron is a unit that takes a set of values as its input, calculates a linear combination of these values and outputs 1 or -1, depending on some threshold ($-w_0$). Figure 6.3 shows a perceptron with input values x_i which are weighted depending on some real valued constant w_i. The output of the node is computed given the input $o(x_1, x_2, \ldots x_n)$ as follows:

$$o(x_1, x_2, \ldots x_n) = \begin{cases} 1 \text{ if } w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n > 0 \\ \\ -1 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{6.3}$$

As one can see in figure 6.3, the threshold can be used as a weight w_0 with a

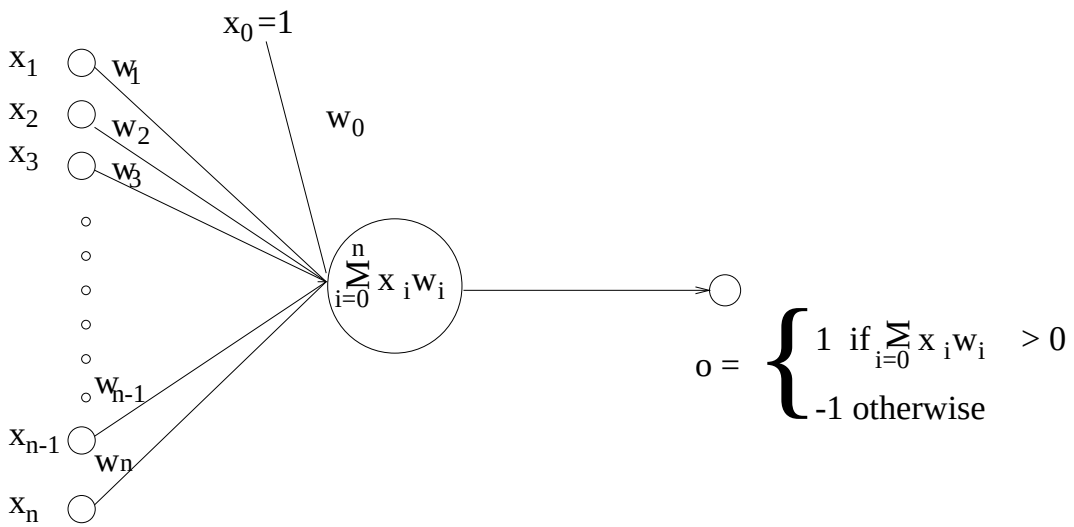


Figure 6.3: A single perceptron

constant activation value of 1 ($x_0 = 1$). This is known as a *bias* and allows one to use the inequality in equation 6.3 as $\sum_{i=0}^{n} w_i x_i > 0$.

6.5.3 Training the Models

Several points have to be taken into consideration when setting up the training of a neural net:

1. The database: how the input and output are encoded
2. The structure of the net
3. The learning algorithm to be employed
4. The training process: percentage of errors, time taken to train, etc.

Each of these shall be discussed in turn.

- ***1. The Database***

The features used to train the net are the same 45 features used to train classification and regression trees as described in section 5.4. These were normalised

between 0 and 1 to restrict the input vector space. The move type of the utterance was encoded into a sequence of 12 binary numbers.

The test and training sets were the same as the CART experiments i.e. set A for training and set C for testing. A quarter of the training set formed a validation set. This set was needed to test the net during training in order to monitor how well it was doing.

- ***2. Net Structure***

The net uses is a three layer network. The input layer consists of 45 units corresponding to the number of features described in section 5.4. The output consists of 12 units, corresponding to the number of move types. In order to perform non-linear decisions, one needs a *hidden layer* of nodes in the net. Experiments were run with different sized hidden layers; the optimal network had a hidden layer consisting of 50 units.

- ***3. The Learning Algorithm***

The learning algorithm used in these experiments is the *backpropagation* algorithm which is used to train multi-layer ANNs and was developed mainly by the PDP group (Rumelhart & McClelland, 1986). This learning algorithm determines the difference between the desired output and the actual output, known as a *cost* or *error function*. The weights of the connections are changed in order to minimise the cost value. This process is known as *gradient descent* and requires the calculation of the gradient of the error function. The weight adjustment that would reduce this error is passed back to the hidden units. Iterative applications of the learning function allow one to find the set of weights that results in the most effective neural net. This process also allows one to determine which of the weights are contributing most to the error.

SNNS has an output function that converts the activation value of the outputs

nodes to a value between 0 and 1. In our case, these figures are used as likelihoods for the 12 moves, given a feature vector. In other words, one takes the node with the highest output value as the most likely move type.

- ***4. The Training Process***

ANNs are similar to CART trees in that they take into account the distribution of moves, that is they output the posterior probability. In order for the ANNs to be used to calculate the most likely move sequence, they need to produce the likelihood of a set of observations, given a type of move $P(I|M)$. As with the CART trees, there are two methods of obtaining this likelihood. The first is to train on a set that has equal numbers of moves. The second involves dividing the output of the ANN by the prior probability of the moves, calculated from the training set (see equation 6.2). The second method produces slightly better results and took fewer training cycles. The results reported in section 6.7 are obtained using this technique.

All patterns are presented once in a random order during one training cycle. Every 10 cycles the net is tested on the validation set and the sum of squared errors (SSE) is calculated. A check is performed and training is terminated if this SSE falls below 630. This allows one to monitor for overtraining. The network achieved optimum classification rate after 50 cycles.

Neural networks are effective at classification given a large feature vector. To enable a network to do this it invents new features using the hidden layer of nodes. Unfortunately, it is very hard to interpret these features. Unlike the classification trees, it is not possible to look "inside" the neural net. The only method of evaluation available is to look at the move recognition accuracy results which are given in the results section 6.7.

6.6 Hidden Markov Models

The previous two sections described methods of modelling intonation using general characteristics of the utterance, for example average F0, amplitude, utterance final F0 excursion etc. They do not really model the sequence of the different types of intonation events, depicted by Pierrehumbert (1980) or Ladd's finite state network (Ladd, 1996) discussed in section 5.2.3. These models also do not attempt to capture characteristics of events in the different parts of the contour (head, nucleus, tail) as described by the British School (Palmer, 1922) discussed in section 5.2.2.

This section presents a method of modelling these different parts of the intonation contour using the different states of a hidden Markov model. HMMs are probabilistic finite state networks and are a common technique in acoustic/phonetic modelling in automatic speech recognition and part-of-speech tagging. A brief overview of the theory behind hidden Markov models is given along with a discussion on whether HMMs are an appropriate method for modelling the intonation contours.

6.6.1 Finite State Networks as Intonation Model

An HMM is formed by adding probabilities to a finite state network. As discussed in section 5.2.3, these networks can be used to model the finite set of sequences of intonation events. Figure 6.4 gives a summary of these FSNs. Figure 6.4a shows Pierrehumbert's intonation grammar giving all the legal tone sequences for English (Pierrehumbert, 1980). Figure 6.4b shows the same information but with descriptive variables associated with states which emit tones of the particular type (e.g. the pitch accent state emits all the pitch accent types). Figure 6.4c shows Ladd's amended version (1996) where nuclear accents are treated differently from pre-nuclear accents. Figure 6.4d shows the British School system of pre-head,

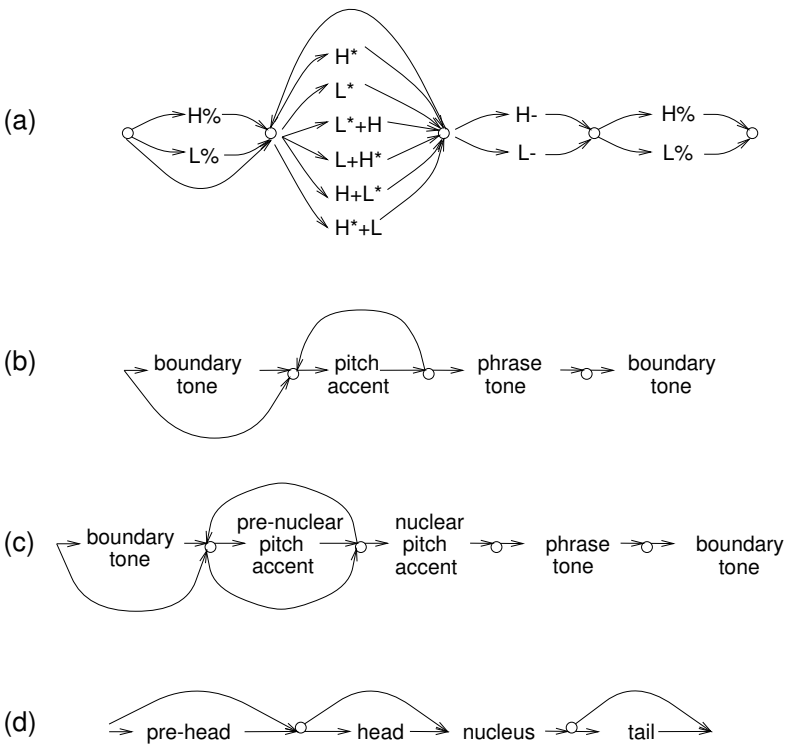


Figure 6.4: Intonation structure represented by finite state networks

head, nucleus and tail, Palmer (1922).

These FSNs are transformed into HMMs by adding two types of probabilities, as illustrated in figure 6.5. *Transition probabilities* (a_{ij}) are added to the arcs between states which give, for example, the likelihood of a contour having or not having a pre-head. *Observation probabilities* ($b_s(o_n)$) are associated with states and specify the likelihood of that state emitting one of the events associated with it. For example, the pitch accent state in figure 6.4b might have a high chance of emitting a common accent such as H* and a much lower chance of emitting a rare accent such as H+L*. The models are known as *hidden* Markov models as the sequence of state transitions is not directly derivable from the observed sequence of accents.

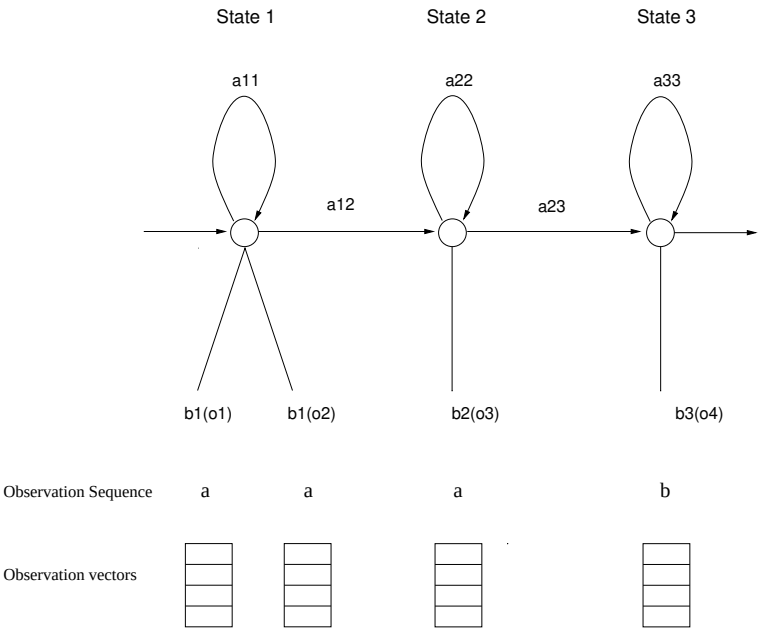


Figure 6.5: A three state, left-to-right HMM

6.6.2 Training the Models

The first stage in training the HMMs is to derive a sequence of observations for each utterance. These observations are intonation events which can either be in terms of discrete labels, such as ToBI or in terms of a set of continuous variables, such as the tilt parameters described in section 5.2.7. The latter is chosen for a number of reasons which are described in chapter 5, but mainly because they are automatically derivable using the method described in Taylor *et al.* (1997).

Recall that this method automatically identifies the intonation events in an utterance[4]. Four tilt parameters (described in section 5.2.7) are used as the observation vector for each event. These parameters are: *start F0, F0 amplitude, event duration* and *tilt. Peak position* is not included for reasons given in chapter 9. Normalisation of these parameters, excluding *tilt*[5], was conducted in an attempt to eliminate speaker specific characteristics. Each observation parameter

[4]The HMMs used for accent detection are separate from the HMMs used for move detection.
[5] *Tilt* is a speaker independent value.

was normalised using the mean and standard deviation of that parameter for a given speaker.

The system is tested on a sequence of automatic identified events. Hand and automatically labelled data are available for training. Although the automatically labelled data may not be as accurate as the hand-labelled data, the models trained on it perform better as they learn the labelling characteristics.

A three state, left-to-right continuous density HMM, as illustrated in figure 6.5, is trained for each of the twelve moves. The HMMs are trained in two stages, *initialisation* and *re-estimation*. Initialisation involves providing crude estimates for the HMM parameters, which are then re-estimated using the standard Baum-Welch algorithm (Baum, 1972). Re-estimation is an unsupervised iterative technique which optimises the maximum likelihood of the models emitting the observations in the training data. The re-estimation process takes several iterations and often after training the states do not emit the same observations as after initialisation. The hidden Markov model tool kit (Young *et al.*, 1996) was used for the training and recognition procedures in these experiments.

A common practice in any recognition system is to retrain the HMMs, increasing the number of component Gaussian mixtures, each with a mean and variance. This provides a more accurate model of the continuous variables' probability distribution, thus improving the recognition results. The best results were obtained using a mixture of two Gaussians.

Various experiments were conducted using HMMs with different transition paths. The most effective was one with no transitions from state 1 to state 3, such as the HMM shown in figure 6.5. The significance of this is discussed in section 6.6.3.

By using a viterbi decoder[6] at run time, the most probable state sequence is

[6]See section 3.4.1 a description of viterbi decoding.

determined, given the observation sequence. A test utterance is passed through each of the 12 models. The likelihood for the intonation of a given utterance is calculated by multiplying the transitional probabilities of the state sequence with the probability of that state emitting the observed intonation event in that state. Unlike the CART or the ANNs, HMMs output the likelihood of the intonation contour given the move type, $P(I|M)$. Therefore no modification is needed regarding the number of each move in the training set.

6.6.3 Discussion

In order to examine exactly how the HMMs model the intonation contour, the relationship between the different types of intonation events and the states that model them was investigated. This was done by using a HTK tool[7] that gives a breakdown of the type of events modelled by each state during recognition. It was observed that prenuclear accents are emitted mostly by state 1, while nuclear accents can be emitted by state 2 or state 3. Boundary tones are only emitted by state 3. The reason why state 3 emits nuclear accents is due to the fact that accents and boundary tones can be combined as a single event if they are close enough together (see section 5.2.7). These findings show that the HMMs are modelling the contour in more or less the same way as the FSNs of the British School and Ladd.

The type of HMM that produces the best results does not allow transitions from state 1 to state 3. Under the assumption that state 2 models the nuclear accents, this also supports the hypothesis that the HMMs model intonation structure in a similar way to the British School and Ladd, represented in 6.4c and 6.4d, respectively. Both of these structures state that the intonation contour must contain a nuclear accent.

HMMs can be used to recognise a sequence of intonation events but they can

[7]By using H2Vite with a trace flag.

also be used to generate such a sequence. In order to gain further insight into the accuracy of the HMMs as models of intonation, they are used to generate intonation contours which are compared to contours of the same move type. This is the topic of the following section.

6.6.4 Resynthesising the Intonation Contour using HMMs

HMMs have a dual function. They can be used to process or generate sequences of intonation events. As discussed in the previous section, they provide the likelihood that an observed sequence of events has been produced by that model. They can also be used to specify the distribution of intonation events for contour generation. Using this latter method, one can test how well the HMMs actually model the intonation by examining the type of contours that they produce.

This statistical intonation contour generation could be used as an alternative to rule based systems in current text-to-speech (TTS) systems (Anderson *et al.*, 1984; Silverman, 1988). General heuristics, such as giving interrogatives a high boundary and declaratives a low boundary, are successful to a certain extent. However, if the conversation is of reasonable length, unnaturalness would be detected by the listener. This unnaturalness is due to the fact that there is not always a one-to-one mapping of intonation to utterance type. For example, a declarative frequently has a low boundary tone, but a high boundary may be realised if there is an element of doubt.

As hidden Markov models are probabilistic finite state grammars, they can capture this variation. However, one has to be very wary of generating intonation contours using statistical techniques due to the unintentional meanings that may be conveyed. Examples of this include placing focus accents on the wrong words or conveying doubt at an inappropriate point in the discourse. For this reason, contour synthesis using HMMs is reported here solely as a tool for examining the effectiveness of the models, not as an alternative to the current intonation

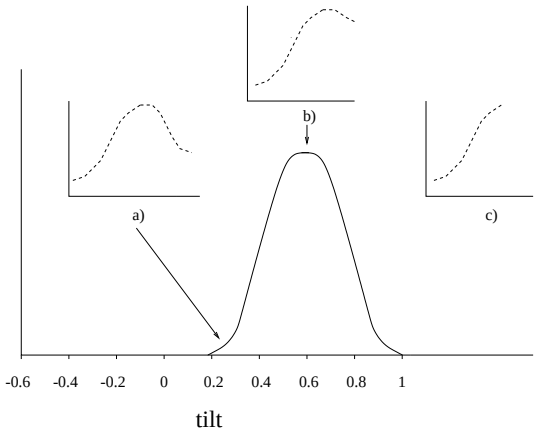


Figure 6.6: Highest weighted Gaussian mixture component for the tilt values of the third state of the *query-yn* HMM

synthesis methods.

- ***Method of Synthesising Contours using HMMs***

Synthesising intonation contours involves computing new tilt parameters for each event in an utterance. These new tilt parameters are calculated depending on the most likely state given the accent's position in the utterance and the observation probability distribution associated with that state.

A simple approach to calculating the values for the four tilt parameters would be to use the mean associated with a specific state. Figure 6.6 shows the highest weighted Gaussian for the tilt value in state 3 of a *query-yes/no* move HMM. In other words, it illustrates the distribution of the shape of boundary tones.

If one just takes the mean (0.6), the HMM will always generate boundaries for yes-no questions that are mostly rising with a slight fall (see figure 6.6b). The model will never predict boundaries with more of a fall which can occur in natural speech. One, therefore, wants to take into consideration the variance of the distribution in order to model the many-to-one mapping of intonation contour to move type. This is achieved by using a random number that has a Gaussian distribution. By doing this, the model will generate some values for tilt that are

rise-fall (nearer 0, figure 6.6a) and some that are complete rise (nearer 1, figure 6.6c), but mostly values around the mean (figure 6.6b).

For each of the four tilt features, a random number is generated[8] with a standard normal distribution, i.e. with a mean of zero and a standard deviation of 1. This number is multiplied by the standard deviation and added to the mean for each parameter associated with a certain state. This is shown in equation 6.4 where R is the Gaussian distributed random variable; x is the new value for that tilt parameter; and μ and σ represent the mean and standard deviation of that parameter, specified by a certain state.

$$x_{tilt} = R * \sigma_{tilt} + \mu_{tilt} \tag{6.4}$$

For simplicity, the Gaussian component that has the highest weighting is used for generation. Recall that each tilt parameter (excluding tilt itself) is normalised to compensate for speaker variation. The predicted values are therefore renormalised using the mean μ^s and standard deviation σ^s of each tilt parameter for a given speaker (s).

$$x^s_{tilt} = \frac{x_{tilt} - \mu^s_{tilt}}{\sigma^s_{tilt}} \tag{6.5}$$

The transitional probabilities determine the state sequence, i.e. the states with higher self-transition probabilities are more likely to be used to produce more accents. The state transitions are determined by a randomly distributed variable. If this random number is above the self-transitional probability then the following event will be generated by the next state. Otherwise, one stays in the same state and generates another random variable. The HMMs are constrained to produce a specific number of intonation events.

[8]This is using the "gauss" program from Entropic (1999).

The contour is generated from the new tilt values by the method described in Taylor (2000).

- ***Synthesis Results***

Figure 6.7 illustrates the types of contours generated by a *yes/no query* HMM. The label files contain the time-aligned word transcriptions and the intonation events. The intonation labels (a, b) indicate the mid points of the automatically detected intonation events. The top contour is the original spoken F0 contour. The second contour is a resynthesised version which is generated from the tilt parameters automatically extracted for each of the events. The bottom two are synthesised using the new parameters generated for each event by the *yes/no query* HMM. Just as human speakers vary boundary tones for such interrogatives so does the HMM. The second of the automatically identified accents is misplaced by approximately 0.2 seconds; it should be aligned with the stressed word "right". This causes the synthesised contours to place an accent on the unstressed syllables "to the". This illustrates that the HMMs are trained on some inaccurate event labels however, as the test set is automatically labelled in the same way, the HMMs would capture these idiosyncrasies for move recognition.

The goal of this work is to produce synthesised contours that reflect variation over a sequence of utterances of the same type. Therefore, standard methods of comparing synthesised contours with the original ones, such as RMSE and correlation, are inappropriate here; for more information see Hermes (1998) and Clark & Dusterhoff (1999).

One can, however, examine the distribution of accents and boundary tones produced by the system over a series of synthesised contours. Boundary tones are chosen as they are labelled in the database as rising or falling (rb/fb) and are likely to contribute to the discrimination of utterance types. Sixty utterances were synthesised using the yes/no query HMM for the sequence of events given

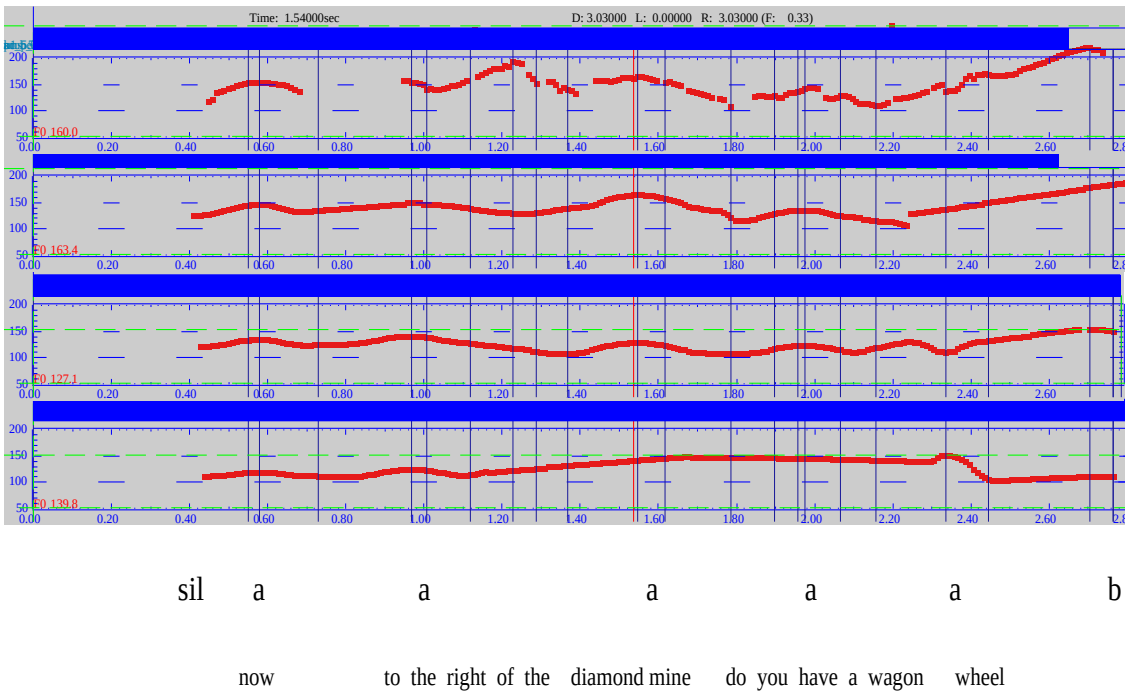


Figure 6.7: Four F0 contours with automatic intonation labelling and words. From top down: original F0; resynthesised F0 from original tilt parameters; two synthesised contours from a yes/no query HMM

.

in figure 6.7. These 60 synthesised contours were then relabelled by hand. In the training data 71% of yes/no questions have a rising boundary; in the synthesised data this figure is 70%. This shows that the HMM does reflect the variation of contours associated with the *query-yn* move type. If the HMMs were inappropriate statistical models or if the method of extracting the tilt features was inaccurate, this distribution of synthesised contours would not have been generated.

Informal listening tests indicate that contours are appropriate to the type of model that produced them.

- ***Discussion***

As described above, the distribution of boundary tone types of the synthesised contours does indicate that HMMs form accurate models of intonation patterns. The state transitions during recognition were examined and it was observed that most of the prenuclear accents are associated with state 1 and most of the nuclear accents with state 2. However, one cannot guarantee that the nuclear accent will be associated with the focus of the sentence or that contrastive stress would be realised appropriately

Final lowering, declination and downstep should be reflected in the distribution of tilt parameters in each state. Again, there is little control over semantic and syntactic influences. For example, in a list scenario each H*+L causes a downstep of the subsequent H*+L accents. This variation in start F0 is captured by the state distribution but as one uses random variables to determine the value of the tilt parameter, one cannot guarantee that the sequence of H*+L accents will decrease in F0 systematically.

Ladd and Johnson (1987) show that the sentence initial F0 height is adjusted according to the syntactic structure of the sentence. In other words, longer sentences have a higher start F0. This is reflected in the values for the mean and standard deviation in the first state of the move HMMs. For example, shorter

move types such as *reply-n* have a lower mean and higher s.d. for start F0 than longer types such as *instruct*. This illustrates that, not only do shorter utterance types start at a lower F0 value, they also have a steeper line of declination. As with downstep, because one is using random variables during synthesis, one cannot be sure that the initial start F0 would be higher that the other accents. This is particularly true with HMMs that have a large start F0 variation, such as *reply-n* and *acknowledge* moves.

Taking these points into account, HMMs cannot be recommended as an alternative to the current contour generation systems found in TTS. This work is presented here, merely as a method for analysing the effectiveness of modelling the intonation contour using HMMs. The main role of these HMMs is to be used in automatic move recognition, the results of which are presented in the following section.

A possible extension of this work would be to perform perception tests using resynthesised contours, such as the examples given above or by integrating the HMM generator into a TTS such as Festival (Black *et al.*, 1996-1999). These tests would look at those cases in which the HMM does not reflect the appropriate intonation contour for a given utterance type. This may lead to redefining the utterance type sets, or re-examining certain aspects of the linguistic representation such as alignment or the *tilt* theory in general.

6.7 Move Detection Results using Intonation

This section presents a comparison of move recognition results using the three different stochastic models presented in the previous sections. These intonation models are combined with the various dialogue models, and the recogniser and language models (as described in chapter 4). This section concentrates on comparing the different methods of statistical modelling. Chapter 7 gives a discussion

	Unigram	**4-gram III &IM**	**4-gram III, IM and REC**
CART	45	49	63
ANNs	45	46	63
HMMs	42	47	64

Table 6.3: Percentage of moves correctly recognised using the intonation models (IM) in conjunction with various dialogue models and the language models and recogniser (REC)

	Total	**Initiating**	**Non-initiating**
HMM, DM, REC	64	56	72
CART, DM, REC	63	55	71
ANN, DM, REC	63	55	72

Table 6.4: Percentage of initiating and non-initiating moves correctly recognised

of the contribution of the intonation models in the system as a whole.

Table 6.3 gives the move recognition results for the three different types of models. The first column gives results using a unigram that takes into consideration the distribution of the moves in the training set. One can see that the HMMs results are slightly lower that the other two methods. There is a general improvement on all the results by using the best dialogue model, 4-gram III (see section 4.5.3). The CART tree has a slightly higher result of 49%. The final column gives results using all three sources of information available for move recognition. These show a significant increase for all models with the HMMs producing a slightly better recognition result of 64%. All these results are well above chance, which is 24%.

Paired t-tests (Iman, 1994) were performed, comparing each model's performance in the different conditions. These tests did not reveal one model to be significantly better at performing move recognition.

In order to discuss the effectiveness of the intonation models, two matrices are given: one using no dialogue model and one using a unigram dialogue model. Results matrices for the system as a whole are given in chapter 7. Table 6.5 gives

	acknowledge	align	check	clarify	explain	instruct	query-w	query-yn	ready	reply-n	reply-w	reply-y	Correct
acknowledge	**223**	0	5	0	6	4	0	0	13	0	0	8	86.1
align	17	**17**	2	0	4	12	0	2	1	0	0	1	30.4
check	20	2	**10**	0	5	24	1	5	0	0	0	0	15.0
clarify	3	0	0	**0**	7	16	0	1	0	0	0	0	0.0
explain	20	0	4	0	**31**	49	0	5	0	0	0	0	28.4
instruct	11	1	4	0	22	**138**	3	15	0	0	0	1	70.8
query-w	7	0	0	0	4	8	**0**	3	1	0	0	1	0.0
query-yn	10	0	10	0	8	39	0	**17**	0	0	0	2	19.8
ready	44	0	0	0	4	0	0	0	**28**	0	0	2	35.9
reply-n	23	0	1	0	4	0	0	0	1	**0**	0	0	0.0
reply-w	8	0	0	0	2	11	0	2	0	0	**0**	0	0.0
reply-y	78	0	3	0	7	2	0	0	8	0	0	**10**	9.3

Table 6.5: Move recognition results for CART tree trained on original data. 44.7% of the moves are correctly classified.

the results matrix produced by the CART tree with a unigram dialogue model which correctly classifies 45% of the test set. The CART tree matrix is presented here as this has one of the highest recognition results using a unigram. One can see by the large number of moves that are recognised as *acknowledge* (464) and *instruct* (303), that the recognition is greatly effected by the the prior probabilities of the moves.

In order to filter out the effect of the prior probabilities, recognition is performed using the CART tree that is trained on equal numbers of moves (see section 6.4.3). The recognition result using this tree is 31.2% and the corresponding matrix is given in table 6.6. The move types that are recognised correctly are more equally distributed; none have 0% accuracy. One can see the effect of the length feature at the top of the CART tree, dividing the data points into short and long utterances. There are confusions between the shorter categories *ready*, *reply-y* and *reply-n* and *acknowledge* types.

	acknowledge	align	check	clarify	explain	instruct	query-w	query-yn	ready	reply-n	reply-w	reply-y	Correct
acknowledge	**126**	4	5	13	3	4	2	0	63	9	7	23	48.6
align	7	**19**	2	5	0	1	3	4	2	2	4	7	33.9
check	9	7	**14**	4	4	1	3	16	2	2	5	0	20.9
clarify	0	0	1	**6**	10	2	1	2	0	0	5	0	22.2
explain	2	2	1	16	**19**	10	19	5	8	3	22	2	17.4
instruct	3	14	10	18	28	**37**	14	47	2	2	20	0	19.0
query-yn	1	2	2	3	3	0	**7**	0	3	0	2	1	29.2
query-w	2	12	9	10	7	3	7	**22**	2	1	7	4	25.6
ready	9	1	0	0	2	1	1	0	**55**	3	3	3	70.5
reply-n	8	0	2	4	1	0	1	0	6	**3**	0	4	10.3
reply-w	3	1	2	1	6	0	4	3	1	1	**1**	0	4.3
reply-y	27	4	4	6	1	1	0	3	23	8	5	**22**	20.4

Table 6.6: Move recognition results for CART tree trained on data of equal move types. 31.2% of the moves are correctly classified.

There is also some misclassification of the longer utterances. In order to see if this misclassification is systematic, the longer move types are grouped into declarative and interrogative types. Specifically, declaratives include *clarify, explain, instruct* and *reply-w* and interrogatives *check, query-yn* and *query-w*. *Align* type moves are classified as "other" as they encompass both declarative and interrogative type utterances. One can see that the *align* model captures this variation in utterance types as both declarative moves (e.g *instruct*) and interrogatives (e.g. *query-yn*) are misclassified as *align*.

The recognition results of the two main utterance types are given in table 6.7. Results are given using the CART tree trained on equal data and the CART tree trained on the original data.

One can see from this table that the tree trained on all the data recognises more declarative utterances than the equal tree. This shows the effect of the large number of *instructs* in the data. The equal data tree misclassifies declaratives as

	IM (all data) %			IM (equal data) %		
	Decl	Interrog	Other	Decl	Interrog	Other
Declaratives	**75**	12	13	**57**	31	12
Interrogatives	50	**26**	24	28	**45**	27

Table 6.7: Percentage of declaratives and interrogative type moves correctly classified by the different types of CART tree

queries depending on their contour. For example, *instruct* moves with a rising contour are often misclassified as *query-yn*. *Explain* moves are frequently misclassified as *query-w* or *reply-w*, all of which typically have have a falling boundary. In addition, utterances that contain an abandoned utterance or a mid-utterance restart tend to be misclassified.

The results in the following chapter show that when the likelihoods based on the recognised words are added, distinguishing between these move types is easier as queries tend to have a fixed syntax. For example, *query-yn* frequently starts with "do you have a" and *query-w* with a wh-word.

The tree trained on equal data is better at classifying interrogatives (45% compared to 26%). Most interrogatives are misclassified as *instructs* by the tree trained on the original data. This is again due to the effect of the prior probability of *instruct* moves.

6.8 Summary

The hypotheses tested in this chapter are firstly, that intonation is indicative of move type, and secondly, that these intonation patterns can be effectively modelled using statistical techniques. Section 6.1 gave a summary of previous attempts at finding intonation contours characteristic of discourse function. Most of these were in terms of rule based models which are problematic for an automatic system as there is not a one-to-one mapping of intonation contours and utterance types. This was the motivation for developing statistical models that can cope

with such variation.

Three methods for modelling intonation statistically were presented. Classification trees are useful as they can take discrete and continuous variables. Unlike neural networks, they do not need a fixed vector space. In other words, the tree can ignore features that are not present, such as the last three accents. HMMs model the intonation from a temporal perspective, with different states representing different parts of the contour. Both CART trees and HMMs can be evaluated to a certain extent by examining the model's internal structure. This has led to the discovery of some interesting intonation phenomenon. One such example is that the classification tree uses the shape of the nuclear accent to decide whether a contour with a falling boundary tone is a yes-no question or another type of question (such as an *align* or *check*).

The three models are very difficult to compare. As well as using different statistical techniques, the models are trained on different sets of data. The CART and ANNs use more general features and can be seen to model the intonation contour from a more holistic perspective. HMMs, on the other hand, take a more autosegmental approach by using the sequence of intonation events characterised by the tilt parameters[9]. This makes a comparison of the models difficult. However, all three models have one important feature in common, namely utterance length. The HMM reflects the length of an utterance based on its transitional probabilities. The CART tree has been shown to use both general features and more theoretically based features such as tilt. This justifies the process of feature extraction based on the sequence of intonation events for contour modelling.

[9]These parameters capture the phonetic properties of an event whereas the autosegmental approach characterises the contour as a string of phonological events.

Chapter 7

System Performance

As discussed in chapter 1, there are two main goals of the work presented in this thesis. The first is to be able to perform automatic move recognition accurately enough to be useful in human-computer interactive systems. The second is to incorporate this move recognition in an automatic speech recognition system to improve word error rate. Automatic move classification results were given in chapter 4, using the recogniser output and language models in conjunction with the different dialogue models. Chapter 6 looked at performing move recognition using statistical models of intonation. Results for move recognition experiments using a combination of these different information sources are given below. All of these experiments are conducted in the overhearer scenario, that is to say the system recognises the sequence of moves solely on the information extracted from the speech (see section 3.4.5).

In order to see if move recognition has the potential to improve word error rate, the speech recogniser is run using hand-transcribed move types. This shows what the error rate would be if the system could obtain 100% move recognition accuracy. The crucial result is when the recogniser is run using the automatically classified move types. This result shows that move recognition rate is accurate enough to improve word error rate. Before discussing the results of the current work, a brief overview of similar studies is given.

7.1 Comparison with Similar Speech Recognition Systems

- *Switchboard Corpus*

The switchboard study (Jurafsky *et al.*, 1997) uses a similar method of move detection for language model selection (see section 2.6 for more details). An utterance type recognition rate of 65% was achieved for their 42 categories, with a baseline of 36%. Regarding word error rate results, using 100% move recognition yields an improvement of 0.9% over the baseline result of 41.2% which is significant ($p < 0.001$). Using the predicted utterance categories results in a slight (non-significant) improvement in word error rate of 0.3 %.

- *Train Projects*

Baggia *et al.* (1997) and Eckert *et al.* (1996) describe the use of utterance type specific language models in train enquiry dialogue systems for Italian and German respectively. Both systems decide on the user's utterance type depending on the system's own utterance, most of which are standard requests. This assumes a degree of cooperation of the speaker which, if violated, results in bad prediction of utterance type, which in turn results in an increase in word error rate. In Baggia *et al.* (1997), using utterance type specific language models reduces the word error rate up to 17% for some utterance types. Eckert *et al.* (1996) use a combination of word and POS bigrams. Using these models results in a significant reduction in word error rate (3.3%) over the general language model.

- *Clarity*

	Information Source	Move type accuracy (%)
A	Baseline	24
B	DM only	35
C	Recogniser output and LM	40
D	Recogniser output and LM and DM	57
E	IM	42
F	IM and DM	47
G	IM, recogniser output and LM and DM	**64**

Table 7.1: Move detection results using various information sources in the overhearer scenario

The Clarity project described in Finke *et al.* (1998) is a speech recognition project using similar data to the switchboard but in Spanish. Three prosodic features (pitch, intensity and speaking rate) were used to train a classification tree for dialogue act classification. Unfortunately, they do not give results for act recognition using prosody. The only dialogue act recognition results they give are in the *transcription scenario*. Specifically, they use the hand-transcribed sequence of words and employ language models and dialogue models to predict the dialogue act type. This result is 48% correct.

7.2 Summary of DCIEM Move Recognition Results

Table 7.1 gives the move recognition results using different combinations of the models described in previous chapters and in Taylor *et al.* (1998b). Using the intonation model trained on all the data yields a move recognition result of 42%,

	acknowledge	align	check	clarify	explain	instruct	query-w	query-yn	ready	reply-n	reply-w	reply-y	Correct
acknowledge	**208**	0	1	0	2	2	0	1	28	0	1	16	80%
align	4	**2**	2	0	2	12	0	4	28	1	1	0	3%
check	11	1	**28**	1	1	3	2	13	1	1	3	2	41%
clarify	0	0	0	**7**	0	17	0	0	0	0	3	0	25%
explain	20	1	9	4	**41**	11	0	11	1	6	5	0	37%
instruct	4	1	1	2	6	**172**	0	2	1	0	3	3	88%
query-w	9	0	4	0	1	2	**4**	2	0	0	0	2	16%
query-yn	6	1	13	0	5	5	1	**54**	0	0	1	0	62%
ready	22	0	0	0	1	3	0	1	**46**	1	0	4	58%
reply-n	4	0	0	0	1	0	0	0	0	**23**	1	0	79%
reply-w	3	0	0	2	5	4	1	0	0	0	**6**	2	26%
reply-y	21	1	0	0	3	3	0	1	0	1	2	**76**	70%

Table 7.2: Confusion matrix for move type classification: 64% move recognition accuracy

which is significantly above chance. The inclusion of intonation models in the system as a whole is justified by the 7% increase in move recognition results (compare D and G). The intonation results given here are those using the HMM, as they yield the best overall result (64%). This figure subdivides into 54% for initiating moves and 80% for non-initiating moves.

Table 7.2 illustrates the distribution of moves correctly analysed using all three models (G). The recognition results are poor for *align* type utterances, which are frequently misrecognised as *ready*. This is most likely due to the similar length of the moves and similar lexical content (mostly "okay"). *Clarify* moves are frequently misclassified as *instruct* as a large portion of *clarify* moves are declaratives with similar discourse functions. Other poorly recognised moves include *query-w* and *reply-w*. This is attributable to the low frequency of these moves, resulting in poor models.

Chapter 3 discussed how the three models are weighted differently to produce

	IM, REC and DM %			**IM %**		
	Decl	Interrog	Other	Decl	Interrog	Other
Declaratives	**75**	8	17	**75**	12	13
Interrogatives	12	**64**	24	50	**26**	24

Table 7.3: Percentage of declaratives and inter correctly classified by CART tree trained on equal data

the optimal move recognition result of 64% (G). These weights were found by using a held out test set as reported in King (1998). He systematically varied the intonation model and recogniser weights, while keeping the dialogue model at a fixed weight of 5. The optimum weights are between 1.5 and 2 for the intonation models and 35 and 40 for the recogniser and language model; see page 140 of King (1998).

In order to examine the misclassification of the longer utterances in more detail, moves are collapsed into declarative and interrogative categories. Recall from section 6.7, declaratives include *explain, instruct, clarify* and *reply-w* and interrogatives *check, query-yn* and *query-w*. Table 7.3 gives the recognition results for these categories using all three modules. These results are compared to those produced by the intonation model trained on all the data given in table 6.7 and repeated here.

The accuracy of recognising declaratives is not improved by using the likelihoods from the dialogue and language models (75%). However, fewer declaratives are misclassified as *query-yn* or *query-w* as these moves have a distinct syntax. There is a significant improvement (40%) in the classification of interrogatives over just using the intonation model. This is likely to be due to the fact that interrogatives contain key words that indicate their discourse function, for example "which, what, do you".

	Information Source	Move Recognition Results%	WER %
A	Baseline	24	24.8
B	DM only	37	26.4
C	Recogniser output and LM	40	24.1
D	Recogniser output and LM and DM	57	24.1
E	IM	42	25.7
F	IM and DM	47	24.7
G	IM, recogniser output and LM	**64**	**23.7**
H	100% move recognition correct	100	23.5

Table 7.4: Move detection and WER results using various information sources in the overhearer scenario

7.3 Word Recognition Results

This section examines whether the move recognition results presented above yield an overall word error rate reduction. Recall that the move type predictions are used to determine the type of language model to be employed during recognition. The method of training the language models is discussed in chapter 4. The word recogniser was run using the move predictions of the different combination of models (A-G). These word error rate results are given in table 7.4, which are also reported in Taylor *et al.* (1998b).

The system achieves a word error rate of 23.5% (H), using the hand-transcribed move types rather than those predicted by the system. This shows a lower word error rate than the baseline result (24.8%) obtained using a general language model trained on all the data. Therefore, the perplexity results of the language

models described in section 4.3.2 do result in a reduction in word error rate. In order for the move type specific language models to be of use, one has to be able to recognise move type with a degree of accuracy. Using the predicted move types (G) to choose the language model yields a recognition result of 23.7%.

Table 7.5 gives the percentage word error rate for utterances grouped into initiating and non-initiating move types. Paired t-tests (Iman, 1994) were performed on the word recognition results for 100% move recognition (H) and the baseline system (A). The reduction of word error rate is highly significant for all the initiating moves ($p < .0005$), but not for non-initiating moves. By combining both results, there is a significant reduction over all the utterance types.

In order to perform t-tests, the utterances tested have to be independent of each other. One cannot claim this due to the nature of the 4-gram dialogue model which uses the types of previous utterances to predict the current one. The data are divided into initiating and non-initiating utterance types. T-tests can be performed between utterances within a set as one can be reasonably sure that utterances of the same initiating/non-initiating type do not affect each other. The decrease in WER of the baseline for initiating moves (from 26% to 24.7%) is significant ($p < 0.001$). The slight increase in non-initiating moves is not significant.

It is interesting to note how well the recogniser would do without the help of dialogue or intonation models (comparing C with A). This result is 24.1%, which is not a significant reduction in WER compared to the baseline[1]. The improvement in results for just the initiating moves is significant ($p < .005$). The baseline result (A) is significantly better ($p < .05$) for non-initiating moves.

[1]The independence assumption is not violated as the dialogue model is not used in these experiments.

Experiment	**Word error rate %**
A: Baseline - General language model	
Overall	24.8
Initiating moves	26.0
Other moves	19.2
C: Move specific language models without dialogue model or intonation	
Overall	24.1
Initiating moves	24.9
Other moves	20.9
G: Move specific language models with automatic move classification	
Overall	23.7
Initiating moves	**24.7**
Other moves	19.3
H: Cheating (100% move classification)	
Overall	23.5
Initiating moves	24.6
Other moves	19.0

Table 7.5: System performance compared with baseline for initiating and non-initiating moves

Information Sources	**Move Recognition Results%**	**Word error rate %**
G: HMM, DM and recogniser output and LM		
Overall	64	23.7
Initiating moves	56	24.7
Other moves	72	**19.3**
G: CART, DM and recogniser output and LM		
Overall	63	**23.6**
Initiating moves	55	**24.55**
Other moves	71	19.56
G: ANN, DM and recogniser output and LM		
Overall	63	23.8
Initiating moves	55	24.75
Other moves	72	19.85

Table 7.6: System performance comparing different intonation modelling techniques

7.4 WER using Different Intonation Models

The results presented in table 7.5 are obtained using the HMMs for intonation modelling. However, the alternative methods, described in chapter 6, produce similar move recognition results (63% compared to 64%). It would therefore be interesting to compare the word recognition results using all three models. These results are given in table 7.6.

This table shows that there is a slight (non-significant) reduction in word error rate using CART trees over HMMs. This improvement is concentrated in the initiating move types. There is no improvement over the HMMs for the non-initiating moves using either the CART or the ANN model.

7.5 Discussion

There are two main questions addressed here. The first is whether the system can perform automatic move recognition with enough accuracy to be useful in human-computer interaction systems. The second is whether the move recognition results are accurate enough to be used in an ASR system to improve WER.

One can see from the results given in table 7.5 that move recognition is more accurate for non-initiating moves than initiating moves. However, the word error rate is better for initiating moves than non-initiating moves, resulting in an important compromise. Initiating moves often contain salient propositional content. Non-initiating moves, on the other hand, are mostly backchannels or replies, where word recognition accuracy is not as essential as knowing the type of response. For example, differentiating "yeah, yes, yep" is not as important as the fact that the utterance is a positive reply to a question. This illustrates how useful move recognition would be in a human-computer interaction system despite the fact that 100% word recognition accuracy is not obtained.

It has been shown that a significant increase in WER can be achieved by using the method presented in this thesis. One can conclude from the results discussed in section 7.3 that intonation models contribute significantly to the move recognition results and the word error rate. The main drawback of the word error results is the small reduction from the baseline to that obtained using 100% move recognition. This gap could be widened by a number of techniques: more sophisticated language modelling techniques; different utterance type sets; and, as always with any recognition task more data. One of these approaches is in the scope of this thesis, namely looking at alternative utterance categorisation. The following section looks at clustering the moves in order to maximise the intonation similarity.

There is an additional source of information that has not been tapped in the

system described so far. This is higher level dialogue information such as a move's position in a game and the game type. The following chapter looks at how this information can be used to improve move recognition accuracy.

7.6 Clustering and Splitting the 12 Moves

One can hypothesise that the inaccuracy of the results presented in the previous section is due to the fact that the 12 moves do not group utterances that are the most intonationally and syntactically similar. Initial experiments presented here involve splitting and merging the 12 moves by hand using heuristics aimed at optimising intonation similarity.

7.6.1 Context Dependent Moves

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.

In their study, *reply-y* and *reply-n* were split into three groups each, depending on the preceding moves: one for those preceded by *align* or *check*, the second for those preceded by *query-yn* and the last for any other move. *Align* and *check* were differentiated from *query-yn* because they have a higher expectancy of a positive answer, whereas *query-yn* moves have a more even distribution of reply types.

A number of possible move sets were experimented with. The set that produced the best results is listed in table 7.7 and was formed in the following way:

- *explain, clarify* and *instruct* were merged as these are mostly declarative

align
check
clarify+explain+instruct
query-w
query-yn
ready
reply-n_prec_by_align+clarify
reply-n_prec_by_query-yn
reply-n_prec_by_other
reply-w
acknowledge+ reply-y_prec_by_align+clarify
reply-y_prec_by_query-yn
reply-y_prec_by_other

Table 7.7: Modified move types

sentences of similar length

- *reply-n* was split into three categories depending on the preceding move: *query-yn; align* or *check*; other
- *reply-y* was split into three types in a similar way to *reply-n*
- *acknowledge* was merged with *reply-y* moves which are preceded by *align* or *check*.
- *align, check query-w, query-yn, ready* and *reply-w* were unchanged

7.6.2 Results of Merging and Splitting

The move and word recognition results were calculated for this new set using various information sources, presented in table 7.8. These were obtained using the same method for the original 12 moves as described in chapter 3.

The overall move detection result is better than the original set (67% compared to 64%), but the figure for chance is higher (28% compared to 24%). In addition the test set perplexity of moves using a unigram is lower than the original test set:

	Information Source	Move Recognition Results%		WER %	
		Original	New	Original	New
A	Baseline	24	28	24.8	24.8
B	DM only	37	45	-	-
C	Recogniser output and LM	40	43	24.1	24.5
D	Recogniser output and LM and DM	57	67	24.1	24.4
E	IM	42	29	-	-
F	IM and DM	47	52	24.7	26.5
G	IM, recogniser output and LM and DM	**64**	**67**	**23.7**	**24.4**
H	100% move recognition correct	100	100	23.5	24.8

Table 7.8: Move detection and WER results using various information sources in the overhearer scenario

6.8 compared to 9.1. This indicates that the new task is easier than the original one.

Using the intonation models alone (E) yields a result of 29% accuracy; this is slightly above the baseline. The intonation models of the original set yield a higher result of 42% on a harder task. This leads one to believe that the new set is not more intonationally similar or that there were not enough data for the less frequent moves, such as the context-dependent replies.

Combining the intonation models with the 4-gram dialogue model (F) increases the result to 52%, compared to 47% of the original moves. The higher dialogue model result (45% compared to 37%) leads one to believe that the new move types follow each other with a higher degree of predictability than the original set.

Adding intonation and dialogue model likelihoods to the recogniser and language models (compare G with C) increases the move recognition (43% to 67%)

but yields similar WER (24.5% to 24.4%). These results are better than than the result using 100% correct move recognition (24.8%). Furthermore, the 100% move recognition result is not lower than the baseline, indicating that the move-specific language models are not better than the general language model. This leads one to believe that the new set does not cluster moves that are syntactically similar, alternatively, it may be that there are not enough data to train language models.

These initial experiments looked at clustering the moves in order to maximise the intonation similarity. This method was unsuccessful from the point of word recognition, as the new move set does not form useful sub-language models. In the following chapter, utterance types are developed that incorporate higher level dialogue information in an attempt to maximise both intonation and syntactic similarity.

Chapter 8

Using Game Information to Improve Move Recognition

8.1 Introduction

This chapter investigates whether higher level discourse information, such as the current discourse goal and goal state, affects the characteristics of utterances of different types. This discourse goal information is captured in Power's (1979) theory of Conversational Game Analysis discussed in detail in chapter 2. Game information is used in this chapter to improve the recognition of move types. Specifically, game type and position of an utterance are used in each of the main stochastic models i.e. dialogue, intonation, and language models.

Game information would be useful in a dialogue model as different move sequences occur in different game positions. For example, a move sequence such as *explain* followed by *acknowledge* is common near the end of a game as the goal of that game is achieved. The type of move can also vary depending on the type of game it is in as each game has a different distribution of moves. For example, in an *instruct* game there is a higher likelihood of finding *acknowledge* moves than in a *query-yn* game where you are more likely to find *reply-y* or *reply-n* moves. The dialogue model can use these regularities to improve move prediction. One motivation for this approach comes from a study by Poesio and Mikheev (1998)

who achieve a 30% increase in move detection by using game information in their dialogue model. An attempt will be made to replicate this study by using game type and game position in the N-gram dialogue model.

Kowtko (1996) shows that the intonation of *acknowledge* moves varies depending on what type of game it is in. She shows that *acknowledges* have a non-falling intonation in non-information seeking games such as *instruct* and *explain* games. *Acknowledges* in the other, information seeking games (such as querying games) tend to have a non-rising intonation.

Intonation may also vary depending on the position of an utterance in a game. For example, if an utterance is game initial, it may be introducing a new goal or topic and have a slightly higher utterance initial F0 contour.

Finally, there may be regularities in syntax corresponding to the game type and position of an utterance. For instance, a *ready* move at the start of a game contains a larger vocabulary than *ready* moves in the rest of the game, as these just tend to consist of "okay". Training language models that take game information into account could improve their predictability.

Although this work follows a similar experimental set-up to the joint work reported in Taylor *et al.* (1998b) and King (1998), the idea was the author's own and the experiments reported in this chapter were conducted by the author[1].

8.2 Automatically Identifying Game information

There are two approaches to using game information for move prediction. One method is to predict game information first and then use this to predict move types. The second method predicts move and game information simultaneously.

The problem with the first of these methods is that game type and position must be predicted with a high degree of accuracy. Initial experiments were per-

[1]With direct correspondence with M. Poesio, S. Isard and S. King.

formed that attempted to recognise game position and game type independently from move type using similar methods to those described in previous chapters. Specifically, intonation, language models and dialogue models were trained for game position and game type recognition separately and simultaneously.

The game dialogue model picks up on the regularities in game type and position. However, as it uses its own predictions it tends to predict the same sequence repeatedly (e.g. "start, middle, end, start, middle, etc.").

Training intonation models on game type alone would assume that utterances have similar intonation if they are in the same game. This is obviously an over-generalisation as a *query-yn* move in a *query-yn* game would not have the same intonation pattern as a *reply* in the same game type. Similarly, utterances of different move type in games of the same type would have very different wording, resulting in poor language models.

It is obvious that more sophisticated methods of game analysis are needed. These may involve: topic spotting (Nakajima & Allen, 1993), cue phrases and discourse marker analysis (Heeman & Allen, 1997), or game boundary intonation studies (Hirschberg & Litman, 1993). The studies mentioned here just look at these features in relation to discourse structure at an utterance level. The development of a sophisticated game annotator is beyond the scope of this thesis.

What will be investigated is whether it is beneficial to model move type features with respect to their game type and position. For example, are game initial *ready* moves more emphatic than game internal ones? In order to do this, a system is trained to identify the product of the move and its game position or type. This system is described in the following sections.

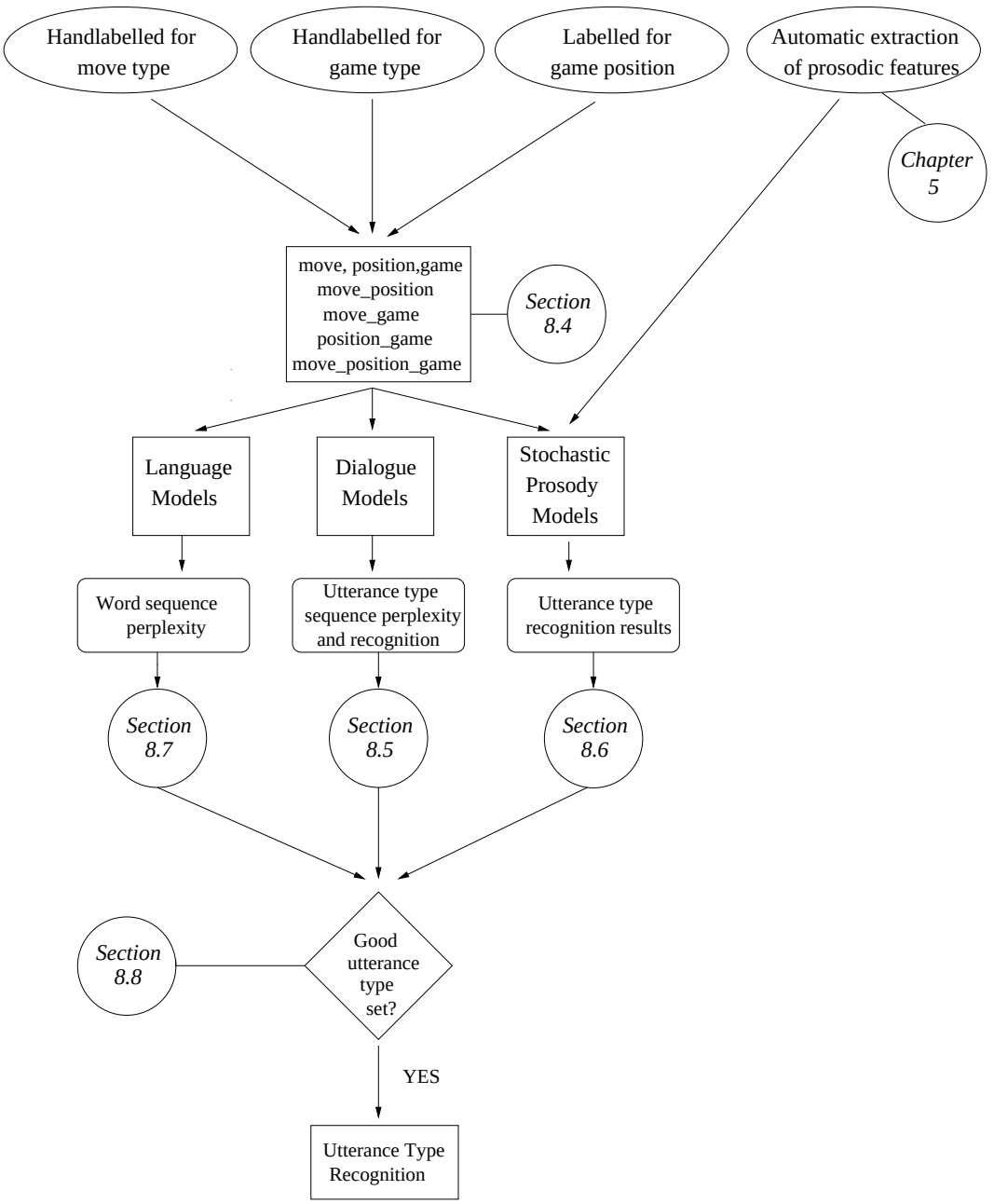


Figure 8.1: Chapter overview

8.3 Chapter Structure

Figure 8.1 gives a schematic representation of the work presented in this chapter. Firstly, the move and game labels are joined to form new sets of utterance types. For example, in order to look at the intonation of move type with respect to position in the game, one uses the move_position set. In order to evaluate the usefulness of these sets, their intonation characteristics are examined, as well as whether they have any syntactic or lexical similarities and whether they occur in a predictable sequence. To do this, intonation models, language models and dialogue models are trained.

Section 8.5 discusses dialogue modelling has two main parts. The first part looks at predicting various combinations of game and move information simultaneously. For example, do utterances in certain game types follow each other with some degree of regularity, such as an explain game final move preceding an *instruct* game initial move? The effectiveness of these models is examined by calculating the perplexity of the category sequence in the same way as explained in chapter 4 for the original move set. The second part of this section looks at improving recognition of the original move set using game information, replicating the findings of Poesio and Mikheev (1998).

Section 8.6 examines whether the new utterance types that include game information form better intonation models than the original set. For example initiating moves at the end of games may have more marked intonation than those at the start. Evaluating intonation models is not straight forward. One can, however, assume that if they improve utterance type recognition then they must model the utterances' discriminatory intonation features to a certain extent.

Finally, section 8.7 looks at whether move specific language models that take game type and position into account reflect syntactic characteristics more accurately than the original set. For example, utterances at the start of games may

contain a new topic or landmark for discussion. The language models are evaluated in the standard way by calculating the test set perplexity of word sequences, using the same method described in section 4.2 for the original move sets.

After examining the three types of models, new utterance types are developed that take into account the move type and game position of an utterance. Section 8.8 presents the results of recognising these new utterance types. The classification of utterances into these new types can be collapsed to the original move types. This yields a significant improvement in the move recognition accuracy presented in chapter 7. Finally, the move types are collapsed further into declarative and interrogative type utterances. These results also show that using game position can improve general utterance type recognition.

8.4 The Data

The game position of each utterance and the game type are hand-labelled for a subset of the DCIEM Map Task corpus. This consists of 25 dialogues which are divided into a training set (B) of 20 dialogues (3726 utterances) and a test set (C) of five dialogues (1061 utterances). None of the test set speakers are in the training set, i.e. the system is speaker independent. The study presented in previous chapters used the larger data set (A) which is labelled for words and moves but not games (see section 3.2 for more details). Therefore, a direct comparison is not possible between results presented in previous chapters and those presented in this chapter.

Another source of data is the Glasgow Map Task corpus. This corpus is obtained using the Map Task set-up described in section 2.4 but is performed by Glaswegian students who are familiar with each other. These data consist of 26,621 utterances and are labelled for games, moves and word transcription but are not labelled for intonation events.

Speaker	Utterance	Move	Position	Game
Giver:	Mike, do you see the start?	align	start	align
Follower:	Yes I do.	reply-y	end	align
Giver:	Do you have a telephone booth just below the start?	query-yn	start	query-yn
Follower:	Yes I do.	reply-y	middle	query-yn
Giver:	Okay.	acknowledge	end	query-yn
Giver:	Go approximately one inch to the left of the telephone booth.	instruct	start	instruct
Follower:	Yes.	acknowledge	middle	instruct

Table 8.1: Data extract including game, position and move type

8.4.1 Data Analysis

The data are analysed in terms of the following categories.

- the original 12 moves
- position in game
- game type

Game type corresponds to the first initiating move in the game. There are therefore six game types corresponding to the six initiating moves given in section 3.2.

Game position is classified as *start*, *middle* or *end*. An alternate set of position types were investigated. This contained an additional move *start_end* which was used in games containing a single move, e.g. an *align* game that contains just an *align* move in between an *instruct* and a *check* game. Initial experiments using a bigram dialogue model on transcribed data showed that including this position type did not improve recognition results. It was therefore discarded and all corresponding moves were labelled as a *start*. Game position was automatically derived from the game boundaries. Table 8.1 gives an extract of data, including move, game type and position labels.

Move type	**#moves**	**Most freq move**	**Baseline%**
position	3	*middle*	43
move	12	*acknowledge*	24
game	8	*instruct*	35
move_position	31	*acknowledge_end*	13
move_game	63	*instruct_instruct*	19
pos_game	18	*middle_instruct*	23
move_pos_game	117	*instruct_middle_instruct*	12

Table 8.2: DCIEM Map Task data statistics for training set B

Move type	**#moves**	**Most freq move**	**Baseline%**
position	3	*middle*	42
move	12	*acknowledge*	22
game	8	*instruct*	29
move_pos	45	*acknowledge_end*	11
move_game	75	*instruct_instruct*	15
position_game	28	*middle_ins*	16
move_pos_game	179	*instruct_middle_instruct*	8

Table 8.3: Glasgow Map Task data statistics

Each utterance can belong to a number of the above categories. Experiments were run on the following combinations:

- move_position (e.g. *align_middle*)
- move_game type (e.g. *align_instruct*)
- position_game type (e.g. *middle_instruct*)
- move_position_game type (e.g. *align_middle_instruct*)

Tables 8.2 and 8.3 give information regarding these categories for the DCIEM and Glasgow Map Task corpus respectively. These include the number of moves for each set, the most frequent move and the corresponding baseline, which is the proportion of the most frequent utterance type in the test set.

One can see from these tables that the Glasgow Map Task corpus has more possible combinations of joint category types. For example, there are 179

Game type	**% of games**	**Av. # of moves per game**
align	6	2.1
check	17	3
explain	14	2.2
instruct	31	4
query-w	7.5	3.3
query-yn	24.5	3

Table 8.4: Distribution and average length of games in training set B

move_position_game types compared to 117 in the DCIEM corpus. This is probably due to the fact that it is a much larger corpus.

The most common move type for both data sets is *acknowledge* and the most common game type is *instruct*. This is to be expected given the nature of the Map Task. Middle position types take up approximately 43% of the move types. This is due to the high number of *instruct* games that have an average of four moves, i.e. an average of two middle moves per game. The game type distribution and average length of the different types of games are given in table 8.4.

Table 8.5 gives the distribution of games and the most frequent initiating move. Not surprisingly, the type of the initiating move and the game type are the same. The table also gives the frequency of the two top non-initiating moves. *Align* and *check* games have more *reply-y* moves than *acknowledges*. *Align* moves frequently contain the phrase "OK?" which is given a positive reply approximately half the time. *Check* is an interrogative usually requiring some kind of confirmation of information or knowledge state and therefore is frequently replied to positively. *Explain* and *instruct* are declarative sentences which are therefore mostly replied to with an *acknowledge*. The last two games, *query-w* and *query-yn*, contain a similar number of *acknowledges* and *replies*. However, the initiating move is most frequently followed by a *reply* rather than an *acknowledge*.

Game type	Init move	Freq	Non-init	Freq	2nd Non-init	Freq
align	*align*	118	*reply-y*	64	*acknowledge*	23
check	*check*	230	*reply-y*	153	*acknowledge*	115
explain	*explain*	290	*acknowledge*	163	*ready*	50
instruct	*instruct*	595	*acknowledge*	382	*ready*	171
query-w	*query-w*	89	*acknowledge*	71	*reply-w*	61
query-yn	*query-yn*	331	*acknowledge*	165	*reply-y*	152

Table 8.5: Frequencies of the most common initiating and the two most common non-initiating moves in various types of games in training set B

8.5 Dialogue Models

In chapter 4, various types of dialogue models were examined, all of which use low level predictors such as sequences of moves or speaker identity. However, the choice of a speech act or utterance type is influenced by other factors, such as current goal and goal state.

As discussed in the previous section, different move types have different distributions depending on the position in the game. For example, 58% of game final moves are *acknowledge* moves. The type of move can also vary depending on the type of game it is in as each game has a different distribution of moves. For example, *acknowledge* moves are more common in an *instruct* game while *check* games are likely to contain more *replies* than *acknowledges*.

This section is divided into two parts. The first looks at whether sequences of moves and game type and position are predictable. The second part looks at replicating Poesio's (1998) experiments by adapting the move dialogue model described in chapter 4 to include game type and position as predictors of move type.

Move type	**Number of moves**	**Unigram perplexity**	**Bigram perplexity**
move	12	9.2	6.2
position	3	2.9	2.56
game	8	5.3	3
move_position	31	18.7	9.8
move_game	63	21.2	8.8
position_game	18	14.3	4.7
move_position_game	117	38.8	17.1

Table 8.6: Dialogue model perplexities for DCIEM corpus

Move type	**Number of moves**	**Unigram perplexity**	**Bigram perplexity**
move	12	9.9	7.2
position	3	2.9	2.6
game	8	5.4	3.2
move_position	45	12	9.4
move_game	75	25	9.4
position_game	28	15.4	5.1
move_position_game	179	47.3	16

Table 8.7: Dialogue model perplexities for Glasgow Corpus

8.5.1 Modelling Game and Move Information Simultaneously

Bigram and unigram dialogue models were trained for the different utterance types and their test set perplexities calculated. These perplexity results are given in tables 8.6 and 8.7. Trigram models were also trained but these did not reduce the perplexity further. In addition, training trigams for a large number of categories such as move_position_game is not possible with the amount of data available.

Game types seem to follow each other with a degree of regularity, as reflected in the reduction of perplexity by using a simple bigram model (from 5.4 to 3.2). For example, *align*, *check* and *explain* games are likely to be followed by an *instruct* game. This would allow for information to be established or checked before giving an instruction. *Query-w* and *query-yn* games are typically followed by *explain* or *instruct* games. If the answer to the query is unsatisfactory, then typically an *explain* game would occur;, otherwise the dialogue continues with an *instruct* game.

Games can be nested, resulting in the need for a more complicated model of dialogue. Initial experiments were conducted that use the binary distinction of nested/non-nested game type. These experiments resulted in a large number of move types, creating sparse data problems. In addition, the games can be deeply nested resulting in abandoned games as the participants forget the initial goal. Even human labellers find it hard to label the end of large games (Carletta *et al.*, 1997). Therefore, trying to distinguish embedded games automatically is impractical at the time of this thesis with the data available.

Chu-Carroll (1998) ran experiments using a dialogue model that only looked at previous dialogue acts at the same level of embeddedness. That is to say, the model would only use previous utterances in the same game. This dialogue model was used to detect the intention of an utterance given its hand-transcribed

syntactic form, i.e. whether it was a question, statement, etc.. Chu-Carroll shows that using this dialogue model does not result in an increase in utterance type recognition over the dialogue model that just looks at the previous utterances regardless of whether they are in the same game or not.

- ***Higher Order Dialogue Modelling for Move_position***

Further Map Task experiments were run to see if using a more complicated N-gram would reduce the perplexity result for the move_position bigram trained on DCIEM corpus (9.8 given in table 8.2). Move_position was chosen for further experiments as it has a manageable number of move types and the bigram model reduces the test set perplexity substantially, indicating that the types follow each other with a degree of predictability. In addition, if the move and position of an utterance is predicted with a certain degree of accuracy then the game type can be inferred from the first few moves in the game.

In order to be able to train an effective dialogue model, there must be a distinctive distribution of move types with respect to their game position. Table 8.8 gives the frequencies of the different moves in different game positions for the training set. One can see that for many move types the distribution is uneven across game positions. The first 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, with the exception of *acknowledge*, have a higher frequency of middle moves than game final moves. 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.

The test set perplexities for various dialogue N-grams are given in table 8.9. One can see that previous move_position utterance types (mp_{i-1}, mp_{i-j}) are better predictors of the current move_position types than just the previous move type.

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 8.8: Move frequencies with respect to game position

Model	Predictors	Perplexity
A	unigram	18.7
B	m_{i-1}	9.78
C	m_{i-j}, s_i, s_{i-1}	9.08
D	m_{i-1}, p_{i-1},s_i	9.06
E	mp_{i-1}, s_i, s_{i-1}	8.55
F	mp_{i-j}, s_i, s_{i-1}	7.6

Table 8.9: Perplexity results for the different dialogue models predicting move_position categories

The 4-gram that reduces the perplexity the most uses the move_position type of the other speaker's previous move (mp_{i-j}) and the current and previous speaker type (Model F). This model is illustrated in figure 8.2 and uses the same type of predictors that were used in the original move dialogue model (King, 1998; Taylor *et al.*, 1998b).

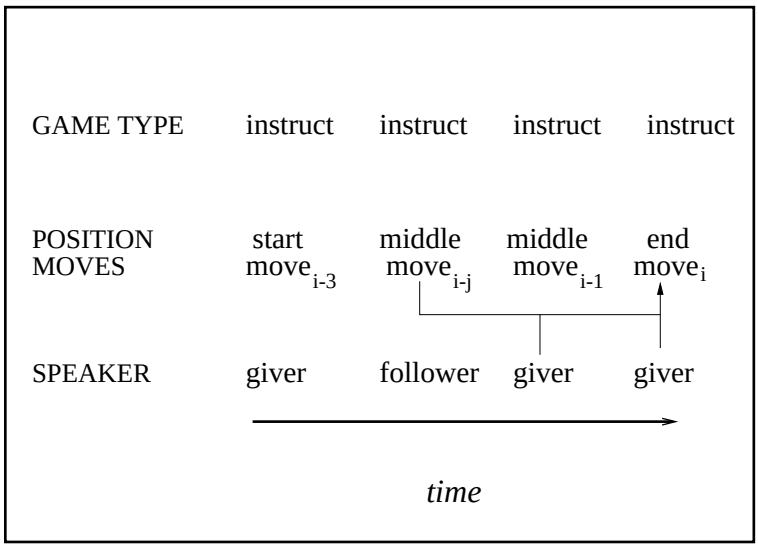


Figure 8.2: Model F for move_position modelling

8.5.2 Comparing ME with N-grams for Modelling Game Structure

As mentioned in the introduction, motivation for this research comes partly from experiments described in Poesio and Mikheev (1998). Their experiments show an improvement in utterance type detection by using game information. The data used for these experiments are the Glasgow Map Task corpus, hand-transcribed for games, moves and words. Instead of using N-grams, they use maximum entropy estimation (Berger *et al.*, 1996) to perform the move discrimination.

Maximum entropy estimation is an alternative way to calculate the probability $P(M|O)$ where O is a set of given observations corresponding to the set of predictors given in table 8.10, e.g. speaker, position in game, etc.. Poesio & Mikheev (1998) report experiments that are conducted in the transcription scenario, discussed in section 3.4.5. That is to say all move and game types used to predict the current move are hand-transcribed, not the model's own prediction, which may contain errors. In their first experiment, they use a simple bigram for move sequences. The result achieved for this method is 38.6% with a baseline

Predictor	**Symbol**
Move type of current move	m_i
Identity of speaker of current move	s_i
Identity of speaker of previous move	s_{i-1}
Move type of previous move	m_{i-1}
Move type of other speaker's last move	m_{i-j}
Position in game of previous move	p_{i-1}
Game type of previous move	g_{i-1}

Table 8.10: Notation of N-gram predictors

Model	**Predictors**	**Perplexity**
I	unigram	9.2
II	m_{i-1}	6.2
III	m_{i-j}, s_i, s_{i-1}	5.1
IV	m_{i-1}, p_{i-1}, g_{i-1}	4.9
V	m_{i-j}, p_{i-1}, g_{i-1}	4.7
VI	m_{i-j}, p_{i-1}, s_i, s_{i-1}	4.7
VII	m_{i-1}, p_{i-1},s_i, s_{i-1}	4.6
VIII	g_{i-1}, p_{i-1}, s_i, s_{i-1}	4.6

Table 8.11: Perplexity results for the different dialogue models for predicting original move types

figure of 21% [2]. The second experiment includes adding a label for game position and game type to each move type. For predicting move types, this method yields a 30% increase to 50.63%. By classifying a separate group of moves, known as "dialogue control" moves (*acknowledge* and *clarify*) and by adding speaker change information, a move recognition rate of 57.2% was obtained.

This section describes a similar study where the hand-labelled position and game type are used to predict the move type of an utterance. Various dialogue models that use combinations of predictors are examined and their test set perplexities for the DCIEM corpus are compared. The full list of predictors is given in table 8.10.

Table 8.11 shows the perplexity of the test set given the dialogue models which

[2]Recall the baseline corresponds to the percentage of the most frequent type.

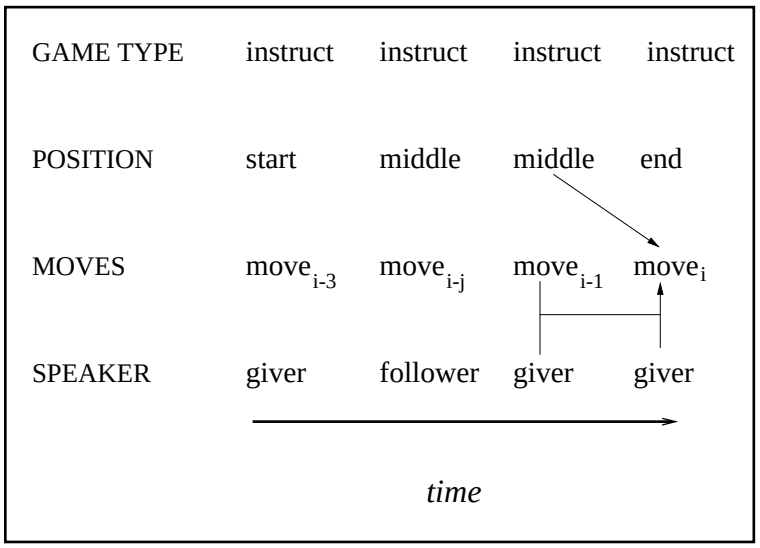


Figure 8.3: Model VII giving 4.6 test set perplexity

use the specified predictors. This table shows that models using game information (i.e. IV to VIII) yield better results than models that just use previous move types and/or speaker information (models I-III). In particular model VII and VIII reduce the perplexity of the source the most. These models use the speaker identity of the current speaker and the speaker of the previous move and the position of the previous utterance in the game. Model VII uses the previous move type, while model VIII uses the game type of the previous utterance. These models are illustrated in figures 8.3 and 8.4.

The lower perplexity of the new models is reflected in the move recognition results. Table 8.12 gives these results using different levels of information for transcribed data, i.e. the original CART intonation model described in chapter 5 and the output of the recogniser and LMs described in chapter 3. A paired t-test was performed for the different conditions, comparing model VII and model VIII with model III. In each case ($t_{p<0.05} > 2.92$)[3] allowing one to accept the alternative hypothesis, concluding that both model VII and VIII are better models than model III. The best recognition rate (69.1%) is achieved using the intonation

[3]Model VII $t = 2.99$; Model VIII $t = 3.69$.

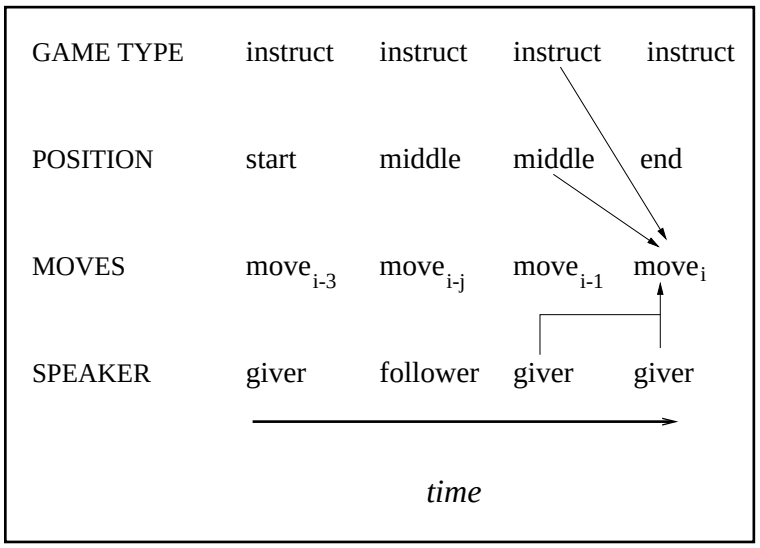


Figure 8.4: Model VIII giving 4.6 test set perplexity

	Model III	**Model VII**	**Model VIII**
DM	52	**55.7**	55
DM, I	54.4	57.6	**58.2**
DM, I,REC	64	**69.1**	68.9

Table 8.12: Percentage of original moves correct using dialogue model (DM), intonation (I) and recogniser output (REC) in the transcription scenario

models, recognition output and dialogue model VII.

To compare methods of dialogue modelling, model IV was trained using both N-gram and maximum entropy estimation methods in the transcription scenario [4]. The model uses the other person's previous move and the position and game type of the previous utterance as illustrated in figure 8.5. Both methods of dialogue modelling produced similar results of 55% for N-grams and 54% for ME.

8.6 Intonation Models

Studies have shown that intonation can be indicative of a change in topic. Nakajima & Allen (1993) show that average F0 at the end and start of an utterance

[4]ME experiments were conducted by Massimo Poesio.

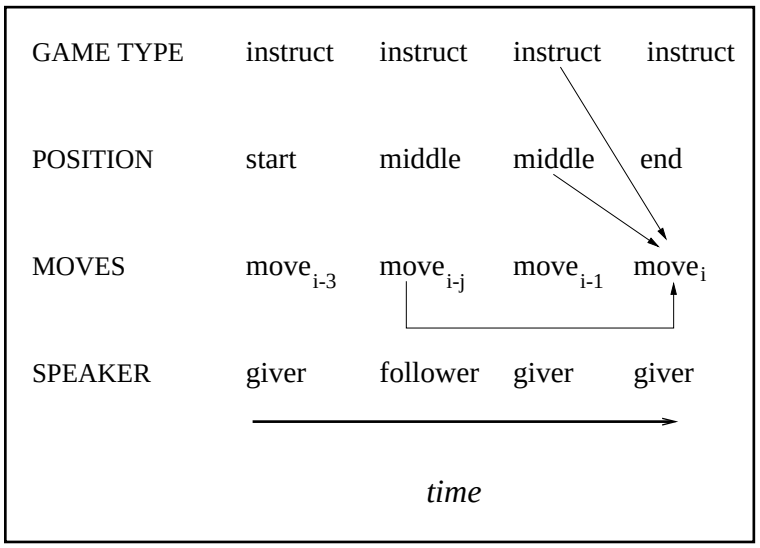


Figure 8.5: Model IV used in ME and 4-gram experiments

	Position	**Position_game**	**Move_pos_game**	**Move_position**
Baseline	43	23	12	35
Bigram	59	47	39	37
Bigram & IM	61	51	42	42

Table 8.13: Utterance type detection using bigrams and intonation models in the transcription scenario

varies depending on whether the utterance is continuing or introducing a new topic (see section 2.10 for more details). 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.

Classification trees are used in the following intonation modelling experiments as this method yields similar move recognition results to HMMs and ANNs but slightly better word recognition results (see section 7.4). One CART tree was trained for each utterance type set (e.g. position) and it was used to classify each utterance into the different categories (e.g. *start, middle, end*)[5].

Table 8.13 gives the results for recognising the different categories of the various

[5]The classification trees were trained on equal numbers of each utterance type.

utterance type sets using a bigram and the respective intonation models. These were initial experiments conducted using a dialogue model that uses the hand-transcribed label of the previous utterance, i.e. the transcription scenario. As discussed above, recognition results are poor using dialogue models to recognise position and position_game in the overhearer scenario. The point of presenting these results is to see whether the intonation models can improve the recognition of the various types of utterances. This is in fact the case, as one can see an improvement by using the likelihoods of the intonation models over using the dialogue model alone. One can infer from this that the CART trees must be able to discriminate utterances using their intonation features to a certain extent.

The intonation model for position does yield a slight improvement. However, there is a greater increase for combined move and position prediction. This suggests that the intonation of a contour is influenced not only by its game position, but also by its move type. For example, an *acknowledge* may have a more declarative type intonation contour, i.e. falling more rapidly, if it occurs at the end of a game than in the middle, emphasising finality. Results for recognising move_position in the overhearer scenario are given in section 8.8.4.

8.7 Language Models

This section looks at whether utterances have distinctive lexical characteristics depending on where they are in a game and the type of game they are in. This is determined by calculating the perplexity of the test set using the appropriate utterance type-specific language model. This perplexity is compared to that calculated using the general language model and the original move-specific language models. If the perplexity is lower, then the sub-language models capture characteristics of the utterances that are missed by the general model. For further details on language modelling and perplexity see section 4.2.

LM type set	Number of models	Perplexity
general	1	27.6
original moves	12	27.2
position	3	30.1
position_game	18	32.4
move_position	31	32.4

Table 8.14: Perplexity results for the test set using the various sets of language models trained on set B

Table 8.14 gives the test set perplexity using the different sets of language models. The last two sets, position_game and move_position, have the highest perplexity. However, one has to take into account the amount of data used to train the specific language models. The higher the number of move types, the less data there is for each language model. Language modelling using the move_position set seems to have potential as the perplexity is only 5.2 higher that the original move set but is used to recognise 19 more utterance types. The position_game language model set, on the other hand, is less effective as it contains only 4 more types than the original set. Modelling word sequences based solely on game type and position is not optimal; for example, *instruct* and *ready* moves that start an *instruct* game will have a quite different syntax.

One can see that using the position language models does not reduce the perplexity of the test set, despite having a large amount of data for training each model. This illustrates, as discussed earlier, that classifying utterances solely on the basis of game position does not group utterances that are syntactically similar.

8.7.1 Smoothing Move_position Language Models

In order to compensate for words that may occur in the test set but not in the training set for each move type, the sub-language models are smoothed with the general language model. This allows the models to cover a larger vocabulary while still capturing the characteristics of utterances of a certain type.

	Original moves	**Position**	**Game_position**	**Move_position**
general	27.6	27.6	27.6	27.6
move specific	27.2	30.2	32.4	32.4
smoothed	27.2	28.3	24.9	27.7
best choice	23.8	27.6	24.6	**23.6**

Table 8.15: LM perplexity results for whole test set using different types of language models for the different utterance type sets

The method for language model smoothing presented here is similar to that described in King (1998) for the original twelve move-specific language models. He calculates the perplexity of the test set using a general language model, move-specific language models and smoothed move type-specific models. For each utterance the model that has the lowest perplexity is chosen; this process is called "best choice move-specific" by King. Section 4.2 gives a more detailed account of the method of training and smoothing language models.

Table 8.15 gives the perplexity for the test set using the various sets of utterance type-specific language models. Position type utterances have the highest "best-choice" figure, indicating that they would not be of much use for word recognition. The best result is obtained using the move_position utterance type sets which is just below the perplexity result calculated using the original move set (23.6).

Recall from section 4.2 that the smoothed models are achieved by combining the general language model with the move specific models using a set of weights. If the model is weighted more towards the move-specific model then one can assume that the sub-language model captures characteristic syntactic information. The smoothing weights for move_position are given in table 8.16, where 1 is total weighting of the utterance type-specific language model and 0 is total weighting of the general language model. This table indicates whether dividing the move type into *start, middle* and *end* provides better language models. One can see that the shorter original moves are weighted more towards the move type-specific language

model, such as *acknowledge*, *reply-n* and *reply-y*. *Reply-y* and *ready* make better language models if divided into the different positions. The *ready_middle/end* moves contain mostly "okay", whereas *ready_start* ones generally have a wider vocabulary. This is reflected in the weights where *ready_middle* is more heavily weighted towards the move-specific language model.

For longer move types such as *explain, clarify, align, check* and *query-w*, the word sequences are more varied and the more data available the better the language model. Hence, the general language model is more heavily weighted. Dividing these moves into the different positions results in more weighting towards the general model as there is less data to train the sub-language models.

8.8 Modifying the Move_position Utterance Type Set

Given the above discussion, it looks worth while trying to merge some of the move_position categories before performing move and word recognition. Clustering of the move_position categories was based on the language model weights given in table 8.16, data sparsity and knowledge of intonation similarity.

The most promising utterance type set, which will be called move_position set 2, contains 19 categories. 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.

The *end* and *middle* moves are combined for the following move types: *instruct*, *query-w*, *query-yn, reply-w* and *ready*. This is motivated by the lack of data of game final moves of these types. This results in poor language models which are weighted towards the general language model during smoothing (these weights in the final column are in bold in figure 8.16). *Align, check*, *clarify*, and *explain* are used as categories regardless of position as they are longer and have a more varied

Original move	Weight	Position	Weight
acknowledge	0.8	end	0.8
		middle	0.8
align	0.3	end	0.4
		middle	0.2
		start	0.3
check	0.2	end	0.0
		middle	0.05
		start	0.25
clarify	0.3	end	0.2
		middle	0.3
explain	0.5	end	0.2
		middle	0.3
		start	0.4
instruct	0.6	end	**0.2**
		middle	0.5
		start	0.4
query-w	0.5	end	**0.0**
		middle	0.4
		start	0.5
query-yn	0.5	end	**0.1**
		middle	0.5
		start	0.5
ready	0.6	end	**0.6**
		middle	1.0
		start	0.6
reply-n	0.8	end	1.0
		middle	0.8
reply-w	0.4	end	**0.1**
		middle	0.4
reply-y	0.7	end	0.5
		middle	0.7

Table 8.16: Smoothing weights towards the move_position and move specific language models

	Original moves	**Move_position**	**Set 2**
number of moves	12	31	19
set B general	27.6	27.6	27.6
move specific	27.2	32.4	**29.7**
smoothed	27.2	27.7	27.9
best choice	23.8	23.6	23.9

Table 8.17: Perplexity results for the test set trained on set B, smoothed with set B general

syntax.

This method of clustering results in a decrease in perplexity using the utterance type-specific language models over the unclustered set (compare 29.7 with 32.4 in table 8.17). This is still slightly higher than the original set (27.2), but one has to remember that there are a larger number of moves with less data to train on (19 compared to 12).

8.8.1 Language Models for Move_position Set 2

In order to see if this new set would be useful for word recognition, the "best-choice" figure was calculated (given in table 8.17) and compared to the original move and move_position language models. One can see that there is little difference in perplexity between these sets. One can infer from this that the moves that were merged have a reasonably similar lexical distribution as no information is lost.

The smoothing weights for the new move_position set are given in table 8.18. Comparing the figures in bold with table 8.16 shows that joining the middle and end moves for *instruct, query-w, query-yn, reply-w* and *ready* creates a better language model than having separate models. However, compared to the original, undivided moves, only *ready* and *query-yn* perform as well as or better than the original language models. Again it may be the case that the original moves do better because they have more data to train on.

Original move	**Weight**	**Position**	**Weight**
acknowledge	0.8	end	0.8
		middle	0.8
align	0.3	all	0.3
check	0.2	all	0.2
clarify	0.3	all	0.3
explain	0.3	all	0.3
instruct	0.6	middle	**0.5**
		start	0.4
query-w	0.5	middle	**0.4**
		start	0.5
query-yn	0.5	middle	**0.5**
		start	0.5
ready	0.6	middle	**1.0**
		start	0.6
reply-n	0.8	end	1.0
		middle	0.8
reply-w	0.4	all	0.4
reply-y	0.7	end	0.5
		middle	0.7

Table 8.18: Smoothing weights towards set 2 and move specific language models

Model	Predictors	Perplexity
A	unigram	14
B	m_{i-1}	8.3
C	m_{i-1}, m_{i-2}	9.7
D	$mp2_{i-1}$, s_i, s_{i-1}	6.9
E	$mp2_{i-j}$, s_i, s_{i-1}	4.5

Table 8.19: Perplexity results for the different dialogue model

	Original moves %	Set 2 %
Baseline	24	13
IM	45	30.4
4-gram	37	25
4-gram & Intonation	47	37

Table 8.20: Recognition results for move and set 2 using Model E and intonation models in the overhearer scenario

8.8.2 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 8.19. One can see a reduction from the unigram perplexity by using a bigram (compare model A with B). No more information is gained by using trigrams (C).

As shown in previous dialogue modelling experiments, speaker identities are good predictors of move and position types. As with the original 12 move types the best perplexity is achieved by using the other person's previous set 2 move type ($mp2_{i-j}$) and speaker identities (model E). This results in a perplexity of 4.5 which is lower than the test set perplexity (5.1) using a similar dialogue model for the original move set, despite the new set having a higher number of moves (19 compared to 12).

8.8.3 Intonation Models for Move_position Set 2

Intonation models were trained to model the characteristics of the new move_position set. Experiments were conducted that performed move recognition in the over-

hearer scenario using intonation models in conjunction with dialogue model E defined in table 8.19. Using the intonation model which takes into account prior likelihoods, achieves a recognition rate of 30%, which is significantly higher than the baseline. Adding the likelihoods of the intonation model to the 4-gram dialogue model E results in an increase of 12% move recognition accuracy.

The matrix given in table 8.21 gives the breakdown of the utterance type recognition performed by the intonation model trained on all the data. One can see a similar division between the short and long move types observed with the recognition performed on the original moves, discussed in section 6.7. For example, the short moves (e.g. *acknowledge, replies, ready*) are often misclassified as *acknowledge_end* or *ready_start*. Short utterances with a rising boundary are typically classified as *ready_start* and ones with a falling boundary as *acknowledge_end*. This pattern is also observed using the tree trained on equal numbers of moves, which has a recognition rate of 16%. Therefore, it is not necessarily the prior probabilities that cause this misclassification but mostly the use of the length feature in the tree.

Regarding the longer utterances, these are again split into two categories: declaratives (*clarify, explain, instruct_inter, instruct_start, reply-w*) and interrogatives (*check, query-yn_inter, query-yn_start, query-w_inter, query_start*). Table 8.22 gives a breakdown of the tree's classification in terms of these main utterance types.

There is no improvement in the classification of declaratives using the new move set. However, the new tree does perform better at recognising interrogative type sentences (compare 32% with 26%). As with the original intonation model, many of the *query* moves are misrecognised as *instruct* type moves, which are the most frequent of the longer move types. This reflects the effect of prior probabilities on the classification tree.

	acknowledge_end	acknowledge_inter	align	check	clarify	explain	instruct_inter	instruct_start	query-w_inter	query-w_start	query-yn_inter	query-yn_start	ready_end	ready_inter	ready_start	reply-n_end	reply-n_inter	reply-w	reply-y_end	reply-y_inter	Correct
acknowledge_end	**80**	14	1	4	0	8	3	0	0	0	0	0	0	0	18	0	0	0	9	0	58.394 %
acknowledge_inter	61	**33**	2	1	0	4	3	0	0	0	0	0	0	0	14	0	0	0	4	0	27.049 %
align	9	2	**17**	2	0	5	9	0	0	0	0	6	0	0	6	0	0	0	0	0	30.4 %
check	9	1	5	**14**	0	8	12	0	0	0	0	11	0	0	5	0	0	0	2	0	20.9 %
clarify	3	0	0	1	**0**	6	11	0	0	0	0	0	0	0	3	0	0	0	0	0	0.0 %
explain	9	0	0	3	0	**44**	34	0	0	0	0	1	0	0	14	0	0	0	0	0	41.9 %
instruct_inter	2	0	3	4	0	31	**79**	0	0	0	0	10	0	0	6	0	0	0	2	0	57.7 %
instruct_start	1	0	1	2	0	9	38	**0**	0	0	0	7	0	0	5	0	0	0	0	0	0.0 %
query-w_inter	0	0	1	0	0	1	1	0	**0**	0	0	0	0	0	2	0	0	0	0	0	0.0 %
query-w_start	4	0	0	3	0	4	3	0	0	**0**	0	1	0	0	2	0	0	0	2	0	0.0 %
query-yn_inter	4	0	0	5	0	0	9	0	0	0	**0**	1	0	0	1	0	0	0	0	0	0.0 %
query-yn_start	4	0	2	11	0	10	26	0	0	0	0	**11**	0	0	2	0	0	0	0	0	16.7 %
ready_end	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0.0%
ready_inter	4	1	0	0	0	1	1	0	0	0	0	0	0	**0**	17	0	0	0	0	0	0.0 %
ready_start	16	1	0	1	0	1	0	0	0	0	0	0	0	0	**33**	0	0	0	2	0	61.1 %
reply-n_end	3	0	0	0	0	1	0	0	0	0	0	0	0	0	4	**0**	0	0	0	0	0.0 %
reply-n_inter	8	3	1	2	0	0	1	0	0	0	0	0	0	0	3	0	**0**	0	3	0	0.0 %
reply-w	5	0	0	1	0	5	10	0	0	0	0	2	0	0	2	0	0	**0**	0	0	0.0 %
reply-y_end	21	6	1	2	0	6	0	0	0	0	0	0	0	0	3	0	0	0	**11**	0	22.0 %
reply-y_inter	21	4	2	2	0	6	3	0	0	0	0	1	0	0	12	0	0	0	7	**0**	0.0 %

Table 8.21: Confusion matrix for move type classification: intonation 30% recognition accuracy

	IM using position %			**IM using original moves %**		
	Decl	Interrog	Other	Decl	Interrog	Other
Declaratives	**75**	9	16	**75**	12	13
Interrogatives	42	**32**	26	50	**26**	24

Table 8.22: Percentage of declaratives and interrogative type moves correctly classified by CART tree trained on all data

8.8.4 Move_position Set 2 Recognition Results

Table 8.23 gives the utterance type recognition results for the new move set 2 and the original move types using the various information sources in the overhearer scenario. In general, the new set has a lower move recognition rate. These results, however, are not directly comparable as recognising the new set of utterance types is a harder task than the original which is reflected in the lower baseline result (13%). However, all of the move_position recognition figures are higher with respect to this baseline than the original move recognition results are to the baseline of 24%. For example, using the dialogue model and the recogniser output (D) yields a move recognition accuracy of 57%, which is 2.4 times the original baseline, whereas the result of 47% for move_position is 3.6 times the baseline for move_position.

8.8.5 Word Recognition using Move_position set 2

After performing move_position recognition, the move_position language models are used for word recognition. It is difficult to compare these word error results to those reported in Taylor *et al.* (1998b) as the amount of data available for training the move_position language models is much less.

One can, however, compare the baseline result using the general model trained on set B with the results using the set B sub-language models. For the utterance specific language models to be of any use the 100% recognition figure must be

	Information Source	Type Recognition Results%	
		Original	New
A	Baseline	24	13
B	DM only	35	25
C	Recogniser output and LM	40	26
D	Recogniser output and LM and DM	57	47
E	IM	42	30
F	IM and DM	47	37
G	IM, recogniser output and LM and DM	**64**	**49**

Table 8.23: Move detection results using various information sources in the overhearer scenario

lower than the baseline. Otherwise, there is no use in trying to predict the move type as it would be more beneficial to use the general model for all utterances. The baseline word error rate using words in data set B is 26.1 where the 100% move_position result is 27.7 (recall lower WER is better). This indicates that the move_position language models as a whole set are not better than the general language model. The word error rate of the system using the recognised sequence move_position categories (of which it gets 49% correct) is 27.6. By system design, this word error rate should be in between the 100% move recognition results (27.1) and the baseline result (26.1) with the 100% move recognition result being lower than the baseline. This is clearly not the case, indicating that this method of utterance type classification is not appropriate for these word recognition experiments.

	Information Source Results%	**Move Recognition**	**Collapsed set2**
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	IM, recogniser output and LM	**64**	**66**

Table 8.24: Move detection accuracy using various information sources in the overhearer scenario

8.8.6 Original Move recognition Results

As discussed above, the whole system classifies 49% of utterances correctly into the move_position categories. If one collapses these move_position labels into the corresponding move type, this results in an increase in the recognition accuracy of original move categories. Results are given in table 8.24 for the original move classification using this method.

With the exception of the dialogue model (B), all the recognition results are increased using the new move types that take position in the game into account. The system as a whole increases the original result by 2% to 66%[6]. A paired t-test was performed in order to compare the two systems using the various information

[6]This increase is non-significant.

sources. The system that uses the move_position categories is significantly better than the one that does not ($p < 0.01$).

The distribution of moves correctly recognised by the whole system is given in the matrix in table 8.25. This table is compared to the matrix in table 7.2 (page 122) produced by the original system. There are several noticeable differences. Firstly, there are fewer *acknowledges* misrecognised as *ready* moves as these rarely occur in the same game position. 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. Less of these move types are confused with *acknowledge* moves that occur in different game positions. More *query-yn* moves are confused with *explains* as the majority of both these move types are game initial.

Surprisingly, there is an increase in *ready* moves that are misclassified as *acknowledges* despite the fact that they rarely occur in the same game position. This misclassification must be due to the fact that the language models have a high weighting and both move types have similar wording, i.e. mostly "okay". Recognising the *replies* using position does not make much of a difference.

Table 8.26 gives the breakdown of these recognition results in terms of initiating and non-initiating moves. One can see that using the move_position set improves the recognition of initiating moves (58% compared to 54%). There is a 3% decrease in the accuracy of non-initiating moves. The position of initiating moves, by their nature, is more predictable than non-initiating ones. This aids the recognition of initiating move_position utterance types and therefore initiating move types.

	acknowledge	align	check	clarify	explain	instruct	query-w	query-yn	ready	reply-n	reply-w	reply-y	Correct %	Original %
acknow.	**232**	1	0	0	1	0	0	1	13	0	1	10	**89.6**	80
align	9	**6**	2	1	1	8	1	6	21	0	1	0	**10.7**	3
check	7	1	**30**	0	4	2	1	16	1	1	1	3	**44.8**	41
clarify	0	1	0	**5**	0	14	1	0	0	0	3	0	20.8	**25**
explain	8	2	6	1	**53**	15	2	7	1	3	5	2	**50.5**	37
instruct	1	1	3	1	9	**171**	5	3	2	0	1	3	85.5	**88**
query-w	3	0	1	0	3	1	**10**	2	0	1	1	2	**41.7**	16
query-yn	3	2	11	0	9	5	2	**50**	1	1	1	1	58.1	**62**
ready	32	0	0	0	1	0	0	1	**41**	1	0	2	52.6	**62**
reply-n	1	0	0	0	2	0	0	0	0	**25**	1	0	**86.2**	79
reply-w	4	0	1	0	6	6	1	0	0	0	**7**	0	**28.0**	26
reply-y	24	0	1	0	2	3	0	4	0	1	1	**72**	66.7	**70**

Table 8.25: Confusion matrix for move type classification: 66% move recognition accuracy

8.8.7 Word Error Rate

The word recognition result using the move predictions derived from the move_position labels is the same (23.7%) despite the 2% increase in move recognition accuracy; see table 8.27. The word error rates using the other combinations of information sources also do not result in an increase in word error rate. One should remember, however, that the word error rate is not always a good indication of the performance of a system. This is discussed further in the final chapter.

8.8.8 Declarative and Interrogative Recognition

The moves are collapsed further into interrogative and declarative utterance types using the same method described in section 8.8.3. Figure 8.6 illustrates the recognition results of these categories using the various statistical models. Firstly, one

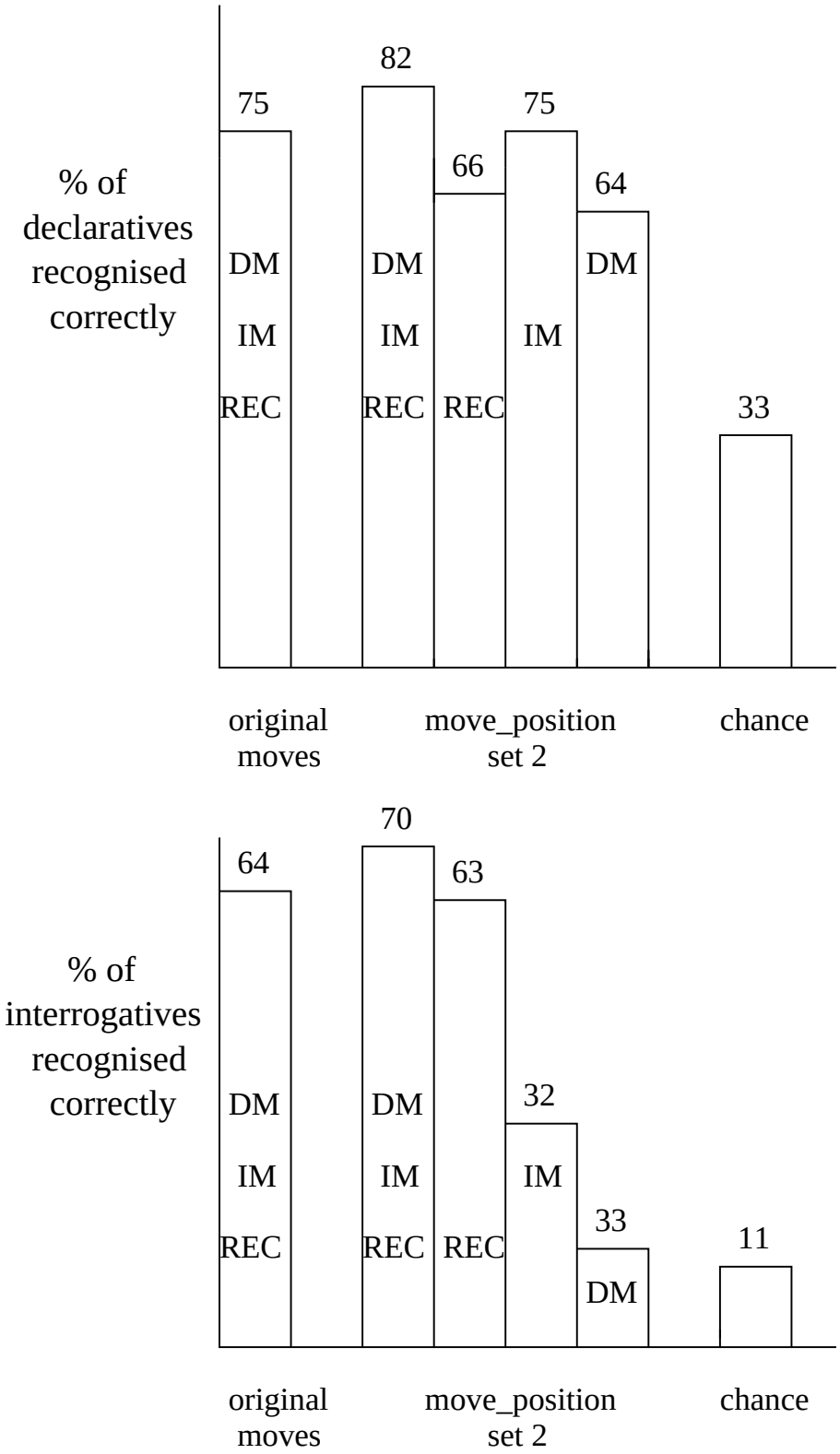


Figure 8.6: Percentage of interrogative and declarative type utterances correctly recognised

Experiment	**Move recognition %**	**WER %**
Cheating		
Overall	100	23.5
Initiating moves	100	24.6
Other moves	100	19.0
Original Move classification		
Overall	64	23.7
Initiating moves	54	24.7
Other moves	80	19.3
Move classification using position		
Overall	66	23.7
Initiating moves	58	24.76
Other moves	73	19.8

Table 8.26: System performance compared with baseline

can see that collapsing the move_position categories predicted by the whole system yields a better recognition result than collapsing the original move set; compare 75% with 82% for declaratives and 64% with 70% for interrogatives.

One can see from figure 8.6 that the intonation models are best at recognising the declarative type sentences (75%) whereas the acoustic and language models perform best at recognising the interrogative type utterances (63%). This is understandable as there is a finite set of words that are used in questions, such as "which, how, etc.". Recognising declaratives, on the other hand, is more difficult as there are no keywords that indicate a declarative type utterance. One can infer from these figures that the intonation of a declarative type utterance is indicative of its discourse function. The intonation of interrogatives, on the other hand, is harder to model.

The dialogue model alone has good declarative recognition (64%) as it assigns the most common move for the follower and the giver each time. These are

	Information Source	Move rec using original method %	WER %	Move rec using position%	WER %
A	Baseline	24	24.8	24	24.8
B	DM only	37	26.4	37	26.6
C	Recogniser output and LM	40	24.1	45	25
D	Recogniser output and LM and DM	57	24.1	**64**	24.6
E	IM	42	25.7	43	27.2
F	IM and DM	47	24.7	50	26
G	IM, recogniser output and LM	**64**	23.7	**66**	23.7

Table 8.27: Move detection and WER results using various information sources in the overhearer scenario

instruct_inter and *acknowledge_inter* respectively. As the model rarely assigns a question type move, the interrogative recognition is poor.

The figure of chance given in figure 8.6 refers to the percentage of declaratives that would be correctly classified if a declarative type was picked 100% of the time. In other words, the test set consists of 33% long declaratives, 22% interrogatives and 45% other type moves.

8.9 Summary

This chapter has looked at the relationship between move type characteristics and the game type and in particular the game position of an utterance. Recognition of move type and game position was performed simultaneously and with a degree of accuracy well above the baseline. These predictions were not directly useful for word recognition. Collapsing the move_position labels to the original 12 move types does result in a significant increase in the recognition accuracy of the system described in the previous chapters. In addition, there is an improvement in the recognition of declarative and interrogative utterance types.

Chapter 9

Automatically Predicting the Syllabic Peak Position

Chapter 5 examined various ways of automatically extracting intonation features based on theoretical assumptions and more holistic properties. A method known as the *tilt* theory described in Taylor (2000) was chosen for automatically identifying intonation events and characterising them in terms of 5 continuous variables, known as the tilt parameters. Recall that these parameters are event shape or *tilt*, start F0, F0 amplitude, duration and peak position.

The three types of statistical intonation model, described in chapter 6, were trained either on the tilt parameters of the whole sequence of events or the last 3 events along with other more general features, e.g. utterance mean F0, s.d. F0. There is, however, one tilt parameter that was omitted from the feature sets, namely *peak position*. This chapter discusses the reasons behind this decision, presents an alternative measure and describes a method of automatically predicting this value.

9.1 Peak Position

Taylor (2000) takes peak position as the distance in time from the start of the utterance to the event peak, as illustrated in figure 5.5 on page 75. This figure

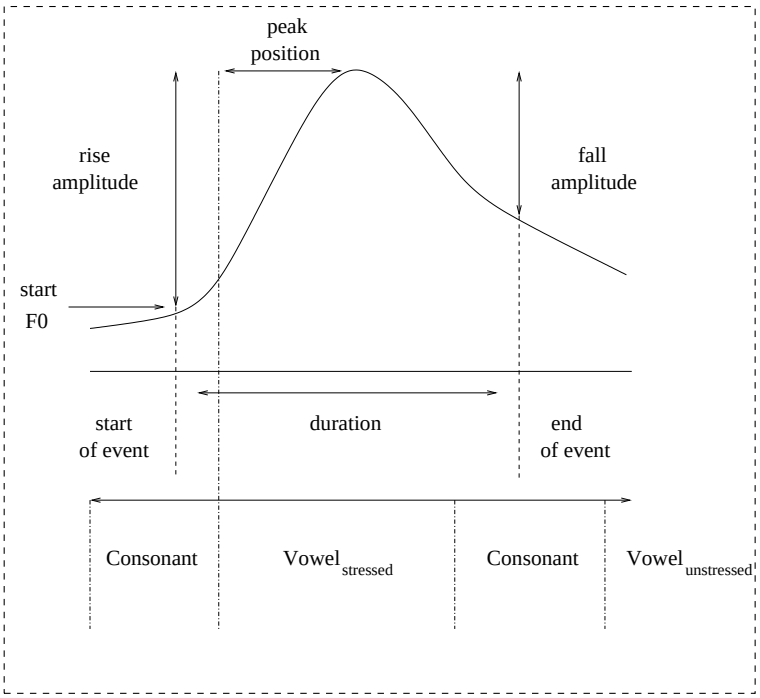


Figure 9.1: Contour schema showing ideal peak position measurement

is necessary for synthesis but is not an intonationally significant feature. Taylor proposes an alternate measure, the *syllabic position parameter*, which is the distance from the peak to the start of the vowel in the stressed syllable, illustrated in figure 9.1. This would provide a parameter that is similar to the other tilt parameters in that it is locally orientated. More importantly, it would capture a distinctive feature of accents associated with accent *alignment*.

Sequences of accents can be linked to the syllables at a phonological level. This is known as *tune-text association*. Figure 9.2 illustrates the segmental and suprasegmental strings and the connections between them. This association of the tiers is non-specific regarding variation of the phonetic *alignment* of the accent with relation to the stressed syllable. The peak of the accent can be late or early in the syllable and in some cases, outside the syllable itself.

There are many factors that may affect the position of the peak of an accent. The study reported in this chapter looks at modelling this peak position using a

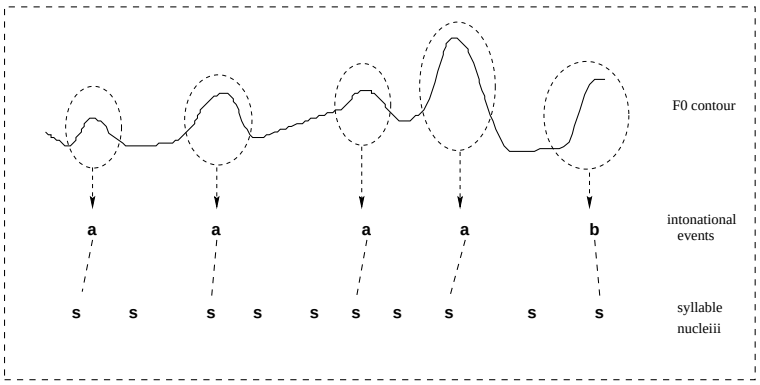


Figure 9.2: Intonation contour with labelled accents and boundary corresponding to the circled pitch excursions. Each accent is linked to a stressed syllable. Source: Taylor (2000).

classification and regression tree (CART). Examining the tree gives insight into which of the other prosodic features influence this position parameter.

Practical applications for a "syllabic peak position predictor" would be to facilitate automatic intonation labelling using features derivable from the signal alone. Peak position could also be used to identify the stressed syllable. Separate triphone HMM models trained on stressed segments would then be used for this syllable during word recognition. This would hopefully improve word recognition error rate. This final research area is, however, outside the field of work presented in this thesis.

Firstly, a review of a number of studies that look at peak position in terms of fixed tonal targets is given. The methodology for training a CART tree to predict syllabic peak position is discussed. Results are presented which show that the trees can accurately predict this peak position value. A discussion follows which examines how the trees predict this value and whether this can contribute to the previous studies examined in the literature review.

9.2 Literature Review

Ladd (1996) highlights the fact that the alignment of the peak in relation to the segmental boundaries is important in distinguishing types of accents. Taking an example using phonologically distinctive ToBI accents, a falling accent can either be labelled as a H* or a H*+L. If it is a H* the peak and therefore the fall occurs later on in the syllable than the H*+L. A measurement of the peak position with respect to the stressed syllable would therefore be a fair representation of peak position.

Bruce's (1977) study of Swedish word accent also exemplifies the discriminatory effect of alignment. He discovered that two distinctive lexical accents are of the same phonological type but they differ in the alignment of the peak with respect to the stressed vowel.

Arvaniti *et al.* (1998) and Ladd *et al.* (1999) developed the notion of *segmental anchoring* which is the constant alignment of the peak and trough of an accent with different parts of the stressed syllable. Segmental anchors can be, for example, the start of the stressed word, syllable or nucleus. Arvaniti *et al.* (1998) show that the peak and the initial low of prenuclear pitch accents in Modern Greek are constantly aligned with the end of the pretonic syllable and the beginning of the post-tonic vowel respectively. They show that the duration of the rise varies depending on the segmental composition of the syllable. So, on short syllables like [dit] in [ro'ditiko], the duration of the rise would be shorter than if the same type of accent is placed on [remv] of [pa'remvasi]. They find that not only is the duration of the accent variable, so is the gradient. Specifically, they show that the F0 values of the maxima and minima remain fixed but the gradient and duration of the rise change to compensate for varying segmental contexts. These findings provide support for the "tonal target" theory which is widely adopted in the literature (Bruce, 1977; Pierrehumbert, 1980; Ladd, 1996). The opposing

theories are known as the "constant slope" theory and "constant duration" theory, where the accent is described by either a characteristic gradient or duration.

Ladd *et al.* (1999) investigate the effect of speech rate on the realisation of rising pitch accents. In a preliminary experiment, they measure the duration and F0 excursion of an accent and hypothesise that increasing speech rate will shorten the duration of the pitch accent but not affect the pitch amplitude. They reject the constant duration theory as there is a strong effect of rate on accent duration. They also reject the constant slope theory as speech rate does not affect the amplitude of the pitch rise.

In a second set of experiments, they measure the distance between two anchor points (i.e. the duration of the test syllable) and the distance between the L and H of the accent. If the duration of an accent is controlled by the anchor points, there should be a strong correlation between the segmental duration of the stressed syllable and the accent duration. This is exactly their findings. They also found no effect of speech rate on pitch excursion height. This goes against the constant slope theory because if the duration is variable and the gradient is fixed then there would be a noticeable affect on pitch amplitude.

Taylor (2000) finds a strong correlation between the tilt figures for F0 amplitude and accent duration which initially seems to be in contrast with the findings of Ladd *et al.* (1999). However, as this correlation figure takes both the rise and fall part of the contour into account, it must be interpreted with care. Taylor does not find a correlation between F0 amplitude and event duration for the separate rise or fall parts of the contour. These findings support the argument against the fixed duration and fixed slope theories.

9.3 Predicting Peak Position

Classification and regression trees (Breiman *et al.*, 1994) were chosen for modelling peak position. Other statistical models such as artificial neural nets (LeCun *et al.*, 1989) were not considered as they are much harder to interpret than decision trees. CART-style decision trees can be used to perform a classification task such as deciding the move type of an utterance, described in previous chapters. A detailed account of the methodology behind classification trees is given in section 6.4.

Decision trees can also also be trained to predict a value given a set of features. This type of tree is called a *regression tree* and is used in the following experiments to predict peak position. For each accent, the regression tree is given a feature vector including variables such as tilt, accent duration, etc.. The tree decides which features are best for predicting the peak position.

9.3.1 The Data

In order to train the statistical model, data are needed that are labelled for phone, word and intonation events aligned to the corresponding stressed syllable. This is problematic for the current study, as the DCIEM data are not labelled for phones and the words are not time aligned. Initial experiments were attempted using forced alignment for phone and word labels. This was run using the HTK toolkit (Young *et al.*, 1996) with the transcribed text and standard phonemic transcriptions. However, this only gives an approximate transcription and alignment, resulting in an inaccurate measurement of peak position.

The data that do comply with these requirements are two sets of read North American English. The first is the Boston radio corpus (BU) containing 2047 utterances read by a female. The second consists of prose on the topic of museum pieces read by a male (known as KED after the speaker's initials). The data

Feature Name	Description
next_peak	distance in seconds to the following accent peak
prev_peak	distance in seconds to the previous accent peak
mean_nrg	mean RMS energy
sd_nrg	standard deviation RMS energy
acc_length	distance in seconds from the peak to the trough
tilt	tilt value of accent
start_F0	start F0 of the accent
F0_amplitude	F0 amplitude of accent
high	position in time of accent peak
low	position in time of accent trough

Table 9.1: List of features used to train the regression tree to predict peak position

contain 786 utterances. The corpora are divided into training and testing sets with three quarters for training and the rest for testing.

9.3.2 Training the Tree

For each intonation event, a set of 10 features is used to train the decision tree to estimate the peak position. These features are given in table 9.1 and are all automatically derivable from the acoustic signal. This set includes the distance in time from the peak of the current accent to the peaks of the next and previous accents. The mean and standard deviation of energy during the accent are also included. The other four tilt parameters are included in the feature set to see if they correlate at all with peak position. All features are normalised to fall between -1 and 1.

Three different types of trees were trained depending on the type of the intonation event: one tree for accents, one for boundaries and one for both types. This was done in order to see if there are any characteristics of boundary tones that could be captured by a separate regression tree. Alternatively, the tree trained on both accents and boundary tones may be able to distinguish the differences

between these types of event.

Events where accents and boundaries occur close together were included in the boundary category as they are linked to the final syllable. Ideally, a different tree should be trained for these events as single boundary tones are typically linked to an unstressed syllable whereas accent+boundary tones are linked to the last stressed syllable in an utterance. Unfortunately the data set would be too small as "ab" events only constitute 5% of the total number of events.

Different experiments were run using alternative segmental anchors other than the start of the stressed vowel. These were start of the word, start of the syllable and end of the vowel. However, the trees were not able to predict these distances with the same degree of accuracy.

Alternative experiments were also run to try and find the distance from the low to a segmental anchor. These results were poor due to the inaccuracy of the automatically labelled data. When calculating the tilt parameters, the program described in Taylor (2000) shifts the accent's position to a certain degree whilst deriving the best fit for the tilt parameter. This results in inaccuracies for the low point of accents.

9.4 Results

The output value of the tree is compared against the real value for each event in the test set. The correlation of the predicted figures and the hand-labelled figures is given in the table below. The root mean square error (RMSE) in seconds is also calculated for these two sets of figures. A correlation above 0.75 and a RMSE below 0.3 show that the tree can predict peak position with a high degree of accuracy.

Tables 9.2 and 9.3 give the correlation results of the various regression trees. The columns give results for the tree trained on the different types of events and

	BU training data		
	Accents	**Boundaries**	**All**
Testing data			
accents	0.84	0.15	**0.85**
boundaries	0.7	0.64	**0.72**
all	0.76	0.12	**0.77**

Table 9.2: Correlation between real peak position and output of the regression tree using the BU corpus for training and testing

the rows are the groups of events that are tested. One can see from the tables that the highest correlation is achieved by using a tree trained on all the data (results are in bold). This indicates that the decision tree takes into account the difference between accent and boundary tones. The best result for accents in the BU corpus is correlation=0.85, RMSE=0.14; and for the KED corpus: correlation=0.73, RMSE=0.2345.

The trees trained specifically for boundaries are worse at estimating the peak position than the other two types of trees that include accents. For example, using the BU corpus, a correlation of 0.64 is obtained using the boundary tree but the tree train on accents alone obtains 0.7 and the tree trained on all events gets 0.73 correlation. These poor results show that it is better to train one tree that can distinguish between the characteristics of accents and boundaries. The poor results for the boundary trees are likely to be due to lack of training data (12% of KED events and 31% of BU events are boundaries). If more data are available, this boundary tree may prove to be a more effective way of modelling peak position.

Tables 9.4 and 9.5 show the results for speaker independent testing. In other words, the tree trained on the BU corpus is used to test the KED corpus and visa versa. One can see a slight reduction in results from table 9.2 to table 9.4 which is to be expected. However, one sees similar and in some cases better results for

	KED training data		
	Accents	**Boundaries**	**All**
Testing data			
accents	0.73	0.5	**0.73**
boundaries	0.44	0.36	**0.5**
all	0.66	0.49	**0.68**

Table 9.3: Correlation between real peak position and output of the regression tree using the KED corpus for training and testing

	KED training data		
	Accents	**Boundaries**	**All**
BU testing data			
accents	0.69	0.34	**0.71**
boundaries	0.41	0	**0.51**
all	0.62	0.1	**0.67**

Table 9.4: Correlation between real peak position and output of the regression tree using the KED corpus for training and BU corpus for testing

the KED speaker independent testing. For example, the correlation using the boundary tree tested on boundaries is 0.36 in table 9.2 but 0.62 in table 9.4. This can be attributed to the fact that BU is a much larger corpus.

Tables 9.6 gives a summary of the best results for speaker independent and speaker dependent testing for accents using trees trained on all events. Table 9.7 gives the results for all the events also using trees trained on both accents and

	BU training data		
	Accents	**Boundaries**	**All**
KED testing data			
accents	**0.73**	0.68	0.71
boundaries	0.55	**0.62**	0.58
all	0.66	**0.69**	0.68

Table 9.5: Correlation between real peak position and output of the regression tree using the BU corpus for training and KED corpus for testing

Data	Speaker dependent testing	Speaker independent testing
BU	0.85	0.71
KED	0.73	0.71

Table 9.6: Correlation results for just accents using accents and boundaries to train the regression tree

Data	Speaker dependent testing	Speaker independent testing
BU	0.77	0.64
KED	0.68	0.68

Table 9.7: Correlation results for accents and boundaries using accents and boundaries to train the regression tree

boundaries. One can see from these tables that there is a strong correlation (up to 0.85) between the predicted value of peak position and the actual value calculated using the hand labelled data.

9.5 Tree Interpretation

In order to examine which features are used the most in calculating the peak position, one can examine decisions made by the regression tree. A measurement of feature usage is calculated which is proportional to the number of times a feature is queried for each training example. Features that are high up in the tree are queried the most. The measurements are normalised for the number of examples in the training set and therefore sum to one for each tree. The feature usage for the BU tree trained on accents and boundaries is given in table 9.8.

One can see from this table that the tilt of an accent is the most discriminatory feature in deciding on peak position. This is also the case for the separate accent and the boundary trees. By examining the tree structure, one can observe that in general, the higher the tilt value the greater the predicted distance from the stressed vowel to the accent peak. This is because the peak occurs nearer the end of the accent if the accent mostly consists of a rise with less fall (see figure 5.4 on

Feature	Usage %
tilt	0.76
mean_nrg	0.25
accent_length	0.12
prev_peak	0.09
sd_nrg	0.08
next_peak	0.02

Table 9.8: Discriminatory features and type usage in peak position prediction

page 74).

If the accent is falling and short and the previous accent is close then the peak position is short. Interestingly, the distance from the previous accent affects the peak position a lot more than the distance to the following peak.

If the tilt value is positive, i.e. the contour is mostly rising, then the peak position depends less on the surrounding accents and more on the mean and standard deviation of the energy. If there is a greater energy mean, then the distance to the peak is longer. In other words, if the accent is more prominent then the peak will occur further away from the start of the stressed vowel. Energy mean must be more indicative of accent prominence that F0 amplitude as this feature is not used by the tree. F0 amplitude is only indicative of prominence given its position in the utterance. The same F0 excursion at the start of the utterance is less prominent than at the end. As tilt is a phonetic model, this information is not available (see section 5.2.7 for more details). Therefore, energy is used as a measure of accent prominence rather than F0 excursion.

9.6 Conclusion

It is difficult to compare this study with that of Ladd *et al.* (1999) because the tilt model does not give a way of accurately examining the rising and falling parts of the contour separately. However, the tree can predict the distance from the peak

to the start of the stressed syllable with a high level of accuracy (correlation is greater than 0.8). The peak position is predictable to a certain extent depending on the shape of the accent, this does not support the constant slope theory.

The distance from the peak to the start of the stressed vowel is also affected by the length of the accent. Separate experiments show a correlation of 0.7 between these two measurements. This contradicts the fixed duration theory as the duration of the accent depends on the segmental string to a certain extent. Level of prominence and the prosodic context of an accent also play a role in the positioning of the peak.

The speaker dependent tree can predict peak position with reasonable accuracy and could be used for automatic prosodic labelling and stressed syllable prediction. Training a speaker dependent regression tree for the DCIEM corpus is outside the scope of this thesis, as the data are not appropriately labelled. This study gives a potential improvement of the tilt system that can be implemented in the future using the DCIEM data.

Chapter 10

Conclusion and Future Directions

As described in the introduction of this thesis, there are two main goals of this work. This chapter discusses whether these goals have been achieved, examines some of the drawbacks of the system and suggests possible areas of further development.

10.1 The Goals

The first main goal is to be able to perform automatic utterance type detection with enough accuracy to be useful in a human computer interaction system. The second goal is to use this utterance type classification to improve speech recognition.

10.1.1 Utterance Type Recognition

Utterance type recognition has been performed with reasonable accuracy (64-66%). This is achieved by training language models, dialogue models and intonation models. These statistical models capture the three main areas of regularity observed across utterances of similar type. In other words, they have similar wording, they follow each other with a degree of regularity and their intonation patterns are indicative of discourse function. It has also been shown that predict-

ing game position and move type simultaneously results in an increase in accuracy for the recognition of the original move types.

10.1.2 Word Recognition

Choosing language models specific to utterance type has been shown to be beneficial in the word recognition process. The method described in this thesis produces a significant increase in word error rate for the initiating move types but not for non-initiating moves. Move recognition accuracy is higher for non-initiating than initiating move types. If one knows the type of the non-initiating utterance then one is not so concerned about getting the word recognition 100% correct. For example, if one knows the utterance is a positive reply to a question, one does not need to differentiate between wordings such as "yeah, yes, yep, etc.".

Although, integrating game information into the system resulted in an increase in move recognition, it did not improve word error rate. One must remember that word recognition is not always a good measure of a system's performance. Spoken dialogue systems tend to have a dialogue manager that performs linguistic and semantic analysis using the recognised words and the utterance type as input. Therefore, the word error rate is not necessarily indicative of the systems ability to extract the propositional content of an utterance. Word error rate is not even a good measure for comparing speech recognition systems. For example, the test set may contain many words which the recogniser finds easy to recognise correctly.

10.2 Areas of further development

10.2.1 Discourse Annotation

One of the main issues in developing a spoken dialogue system is that one is very dependent on the discourse analysis theory adopted. Discussions in previous

chapters have shown the difficulty in developing models that capture the syntactic and intonation similarities of utterances.

The problem of developing an ideal dialogue annotation scheme has been touched upon in this thesis. Firstly, moves were clustered to develop types that were intonationally similar by examining the context of an utterance. In separate experiments, game information was used in the hope of developing a categorisation that grouped utterances both in terms of syntactic and intonation similarity. Neither of these new utterance types produced useful language models for word recognition. However, the game utterance type set was used to improve the recognition of the original move types.

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.

10.2.2 Utterance Type Recognition

The accuracy of automatic move recognition is calculated by comparing the system's output with that of one human labeller. It has been shown (Carletta *et al.*, 1997) that human labellers do not agree 100% of the time (see section 2.4). It would therefore be interesting to calculate the kappa statistic (Carletta, 1996) of the system compared to a number of human labellers. This may present the system in a better light than just calculating the accuracy compared to that of one labeller.

One possible drawback of the system design is that the length feature is used both in the language model and the intonation model. This violates the independence assumption in equation 3.1 on page 42. The length of an utterance would be reflected in the likelihood output of the language model as this is the product of the likelihood of all the recognised words. It is thought that this effect would

be negligible, although further experiments would be needed to prove this.

In order to make the system completely automatic, a move boundary detector would have to be integrated into the system. This could be done simultaneously or separately from the move type detection; see for example Warnke *et al.* (1997). The hand-labelled utterance boundaries were used in this study as detecting them automatically was outside of the scope of this thesis.

10.2.3 Intonation Models

Intonation is the most difficult aspect of discourse to model due its variable nature with relation to discourse function. It has been shown in this thesis that intonation can be modelled using statistical techniques and that these models can contribute to the recognition of utterance types as well as provide a useful investigative tool.

One possible area of investigation is the realisation of intonation with respect to the semantic content of an utterance. For example, a person's intonation may vary depending on whether he/she is talking about a topic which he/she is enthusiastic about, such as a favourite football team. This may not be applicable in the context of Map Tasks performed by the military, but it may be of interest in corpora such as the switchboard data where there is a wider range of topics discussed.

10.2.4 Dialogue Models

Dialogue models could be developed by increasing the number and type of predictors used. New predictors could include extralinguistic features such as eye contact or head movement. A study of the Glasgow Map Task (Anderson *et al.*, 1991), which is coded for these features, shows that the task is completed in fewer moves and 13% fewer words when eye contact is present.

As with all three of the statistical models used in this system, increasing

the amount of training data would improve the performance of dialogue models and allow for higher order N-grams to be developed. Increasing the number of predictors of the current system would require some backing off to cope with combinations that occur in the test set but not in the training set. For example, if a trigram is not in the training set, backing off involves using the likelihood given just the previous move type. However, the dialogue models that perform the best are those that contain mixed predictors and it is not obvious which of these predictors should be backed off. An investigation into the optimal method of backing off mixed predictor N-grams is one area of future development.

10.2.5 Language Models

The main drawback of the system from the point of view of word recognition, is the fact that even if one has perfect move classification, the decrease in word error rate from the baseline is not particularly large (24.8% to 23.5%). The recognition result of the system using the recognised moves falls in between these two figures. Possible ways of increasing this difference are: increase training data; use more sophisticated language modelling techniques; use more syntactically similar utterance types.

The amount of data available to train the language models is of particular importance. The language models trained to recognise the move and position of an utterance were trained on a smaller amount of data than the original set reported in Taylor *et al.* (1998b). Despite this, the move_position models are 2% more effective at recognising the move types after collapsing the categories. This indicates that the method described in chapter 8 shows potential to improve move recognition further if more data labelled with game information are available.

As discussed in chapter 8, there is a large Glaswegian Map Task corpus labelled for words, moves and games which could be used to train the language models. However, there are important differences between these two corpora. These in-

clude differences in dialect, relationship between the participants, approach to the task, sex, age, etc.. Merging these two sources for language modelling would involve smoothing techniques and is one area that could be investigated in the future.

10.2.6 Discourse Markers

In addition to language models, some kind of discourse marker and cue phrase detector could be of use in this system for move and game boundary detection. As the maps have a limited number of landmarks, these phrases could be identified. If an utterance introduces a new landmark then it is likely to be an initiating move and also a start of a game.

Discourse markers, such as "and, then, well etc." are also indicative of the discourse function of an utterance. For example, many *check* moves start with "so". Although language models pick up on this to a certain extent, they do not attached significance to the position of these words in the utterance. For example, the word "well" could either be a landmark or a discourse marker. If it is at the start of an utterance it is likely to be a discourse marker which occurs most often in *explain* moves. In addition, language models do not pick up on discourse marker phrases. For example, many *instruct* moves start with "and then".

In general, initiating moves are more likely to start with discourse markers than non-initiating moves (34% compared to 12%). This is because utterances at the start of a game or adjacency pair have a less certain role and a higher cognitive load than non-initiating utterance types. This is useful for move and word recognition. If a discourse marker is identified at the start of an utterance then one would increase the acoustic weights of the initiating moves. For word recognition, if the previous move is a non-initiating move then higher weights would be placed on the recognition of the discourse marker words.

10.3 Conclusion

In conclusion, the work presented in this thesis shows that it is possible to capture aspects of dialogue using statistical models. These models provide a useful tool for investigating certain phenomena, such as the intonation characteristics of an utterance. This thesis has described a system that uses observations about an utterance and its context to predict the most likely sequence of utterance types. This is useful in spoken language systems, meeting summarisers, data annotation and automatic speech recognition. It is this last application that has been examined in this thesis and it has been shown that integrating an utterance type detector does increase the number of words the system recognises correctly.

Other aspects of speech and dialogue have also been examined. These include experiments involving modelling the peak alignment of an accent. The effect of using high level dialogue information was also examined. This proved to be useful as it improved the utterance type recognition accuracy of the original system.

References

Allen, J. F., Miller, B. W., Ringger, E. K., & Sikorski, T. 1996 (June). Robust understanding in a dialogue system. *In: Proceedings of the 34th Meeting of the Association for Computational Linguists (ACL '96).*

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E. H., Doherty, G. M., Garrod, S. C., Isard, S. D., Kowtko, J. C., McAllister, J. M., Miller, J., Sotillo, C. F., Thompson, H. S., & Weinert, R. 1991. The HCRC Map Task Corpus. *Language and Speech*, **34**(4), 351–366.

Anderson, M. D., Pierrehumbert, J. B., & Liberman, M. Y. 1984. Synthesis by rule of English Intonation patterns. *In: International Conference on Speech and Signal Processing.* IEEE.

Arvaniti, A., Ladd, D. R., & Mennen, I. 1998. Stability of tonal alignment: the case of Greek prenuclear accents. *Journal of Phonetics*, **26**, 3–25.

Austin, J. L. 1962. *How to do things with words.* Oxford University Press.

Baggia, P., Danieli, M., Gerbino, E., Moisa, L. M., & Popovici, C. 1997. Contextual information and Specific Language Models for Spoken Language Understanding. *Pages 51–56 of: Proceedings of SPECOM '97, Cluj-Napoca, Romania.*

Bard, E. G., Sotillo, C., Anderson, A. H., & Taylor, M. M. 1995. The DCIEM Map Task Corpus: Spontaneous Dialogues under Sleep Depriva-

tion and Drug Treatment. *In: Proceedings of the ESCA-NATO Tutorial and Workshop on Speech under Stress, Lisbon.*

Baum, L. E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process. *Inequalities*, **3**, 1–8.

Bennacef, S. K., Néel, F., & Maynard, H. B. 1995 (May). An Oral Dialogue Model based on Speech Acts Categorization. *Pages 237–240 of: Proceedings of the ESCA Workshop on Spoken Dialogue Systems.* Fergus has a copy of this volume; Alan Black has lost his.

Berger, A., Della Pietra, S., & Della Pietra, V. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, **22**(1), 39–72.

Black, A. W., Taylor, P., & Caley, R. 1996-1999. *The Festival Speech Synthesis System system.* Manual and source code avaliable at http://www.cstr.ed.ac.uk/projects/festival.html.

Black, A. W., & Hunt, A. 1996. Generating F0 contours form ToBI labels using linear regression. *Pages 1385–1388 of: Proceedings of ICSLP '96, Philadelphia, Penn*, vol. 3.

Breiman, L., Friedman, J., & Olshen, R. 1994. *Classification and Regression Trees.* Chapman and Hill.

Bruce, G. 1977. *Swedish Word Accents in Sentence Perspective.* Ph.D. thesis, University of Lund.

Bucklow, J., Huber, R., Warnke, V., Batliner, A., & Noeth, E. 1999. Mulit-lingual Prosodic Parsing. *In: ESCA Workshop on Dialogue and Prosody.*

Carletta, J. 1996. Assessing agreement on classification tasks: the kappa statistic. *Pages 249–254 of: Computational Lingusitics*, vol. 22.

Carletta, J., Isard, A., Isard, S., Kowtko, J., A. Newlands, A., Doherty-Sneddon, G., & Anderson, A. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, **23**, 13–31.

Chu-Carroll, J. 1998. A Statistical Model for Discourse Act Recognition in Dialogue Interactions. *Pages 98–105 of: Applying Machine Learning to Discoure Processing.*

Church, K. W., & Gale, W. A. 1991. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language*, **5**, 19–54.

Clark, R. A. J., & Dusterhoff, K. E. 1999. *Objective Methods for Evaluating Synthetic Intonation.* Proceedings of Eurospeech '99.

Core, M. G., & Allen, J. 1997. Coding Diaogues with the DAMSL Annotation Scheme. *In: Working Notes of AAAI Fall Symposium on Communicative Action in Humans and Machines.*

Crystal, D. 1969. *Prosodic Systems and Intonation in English.* Cambridge Studies in Linguistics. Cambridge Universtity Press.

Crystal, D. 1972. The Intonation System of English. *In:* **Bolinger, D.** (ed), *Intonation.* Penguin.

Dusterhoff, K. 2000 (forthcoming). *Synthesizing Fundamental Frequency Using Models Automatically Trained from Data.* Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, U.K.

Eckert, W., Gallwitz, F., & Niemann, H. 1996. Combining stochastic and linguistic language models for recognition of spontaneous speech. *Pages 423–426 of: Proceedings of ICASSP '96*, vol. 1.

Entropic, I. 1999. http://www.entropic.com/.

Finke, M., Lapata, M., Lavie, A., Levin, L., Tomokiyo, L. M., Polzin, T., Ries, T., Waibel, A., & Zechner, K. 1998. Clarity: Inferring Discourse Structure from Speech. *In: Applying Machine Learning to Discourse Processing.*

Fujisaki, H., & Kawai, H. 1982. Modeling the Dynamic Characteristics of Voice Fundamental Frequency with Applications to Analysis and Synthesis of Intonation. *In: Working Group on Intonation, 13th International Congress of Linguists, Tokyo.*

Grice, M., Reyelt, M., Bensmuller, R., Mayer, J., & Batliner, A. 1996. Consistency in transcription and labelling of German intonation with GToBI. *In: Proceedings of ICSLP '96.*

Grosz, B., & Hirschberg, J. 1992. Some intonational characteristics of discourse structure. *Proceedings of ICSLP '92, Banff.*

Grosz, B., & Sidner, C. 1986. Attention, intention and the structure of discourse. *Computational Linguistics*, **12**(3), 172–204.

Gussenhoven, C. 1984. *On the grammar and semantics of sentence accents.* Dordrecht : Foris, 1984.

Heeman, P., & Allen, J. F. 1997. Intonational Boundaries, Speech Repairs and Discouse Markers: Modeling Spoken Dialog. *Proceedings of the 34th Meeting of the Association for Computational Linguists (ACL '96).*

Hermes, D. J. 1998. Measuring the Perceptual Similarity of Pitch Contours. *Journal of Speech Language and Hearing Research*, **41**, 73–82.

Hieronymus, J. 1989. Automatic Sentential Vowel Stress Labelling. *Pages 226–229 of: Proceedings of Eurospeech '89.*

Hinkelman, E. 1990. *Linguistic and Pragmatic Constraints on Utterance Interpretation.* Ph.D. thesis, University of Rochester.

Hirschberg, J., & Litman, D. 1993. Empirical Studies on the Disambiguation of Cue Phrases. *Computational Linguistics*, **19**(3).

Hirschberg, J., & Nakatani, C. 1993. A speech-first model for repair identification in spoken language systems. *Proceedings of Eurospeech '97, Berlin.*

Hockey, B. A., Rossen-Knill, D., Spejewski, B., Stone, M., & Isard, S. 1997 (Sept.). Can you predict responses to yes/no questions? Yes, no and stuff. *Pages 2267–2270 of: Proceedings of Eurospeech '97*, vol. 4.

Houghton, G. 1986. *The Production of Language in Dialogue: A Computational Model.* Ph.D. thesis, The University of Sussex.

Houghton, G., & Isard, S. 1987. Modelling Cognition. John Wiley and Sons Ltd.

Huang, X. D., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F., & Rosenfeld, R. 1993. The Sphinx-II speech recognition system: and overview. *Computer Speech and Language.*

Iman, R. L. 1994. *A data-based approach to statistics: concise version.* Duxbury.

Jekat, S., Klein, A., Maier, E., Maleck, I., Mast, M., & Quantz, J. 1995. *Dialogue Acts in VERBMOBIL.* Tech. rept. 65. Verbmobil.

Jurafsky, D., Bates, R., Coccaro, N., Martin, R., Meteer, M., Ries, K., Shriberg, E., Stolcke, A., Taylor, P. A., & Ess-Dykema, C. V. 1997. Johns Hopkins LVCSR Workshop-97 SWDB Discourse Language Modelling Project Report. *In: CLSP/JHU Summer Workshop on Innovative Techniques for Large Vocabulary Conversational Speech Recognition.* Johns Hopkins University, Baltimore. `http://www.cstr.ed.ac.uk/publications/pending/Taylor_pending_c.s.ps`.

Katz, S. M. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **ASSP-35**(3), 400–401.

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

Kita, K., Fukui, Y., Ngata, M., & Morimot, T. 1996. Automatic acquisition of probabilistic dialogue models. *Pages 196–199 of: Proceedings of ICSLP '96*, vol. 1.

Kowtko, J. C. 1996. *The Function of Intonation in Task Oriented Dialogue.* Ph.D. thesis, University of Edinburgh.

Krippendorff, K. 1980. *Content Analysis: an introduction to its methodology.* Sage Publications.

Ladd, D. R., Faulkner, D., Faulkner, H., & Schepman, A. 1999. Constant segmental anchoring of F0 movements under changes in speech rate. *Journal of the Acoustical Society of America*, **106**, 1543–1554.

Ladd, D. R. 1996. *Intonational Phonology.* Cambridge Studies in Linguistics. Cambridge University Press.

Ladd, D. R., & Johnson, C. 1987. Metrical Factors in the Scaling of Sentence-Initial Accent Peaks. *Phonetica*, **44**, 238–245.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. 1989. BACKBROPAGATION applied to handwritten zip code recognition. *In: Neural Computation*, vol. 1.

Liberman, M. 1975. *The Intonational System of English.* Ph.D. thesis, MIT. Published by Indiana University Linguistics Club.

Liberman, M., & Pierrehumbert, J. 1984. Intonational Invariance under Changes in Pitch Range and Length. *In:* **Aronoff, M., & Oehrle, R. T.** (eds), *Language Sound Structure.* MIT Press.

Maier, E. 1997. *Evaluating a Scheme for Dialogue Annotation.* Tech. rept. 194. Verbmobil.

Mayo, C., Aylett, M., & Ladd, D. 1997 (Sept.). Prosodic Transcription of Glasgow English: An Evaluation Study of GlaToBI. *In: Intonation: Theory, Models and Applications.* ESCA, Athens, Greece.

Nagata, M., & Morimoto, T. 1993. An experimental statistical dialogue model to predict the speech act type of the next utterance. *Pages 83–86 of: Proceedings of the International Symposium of Spoken Dialogue.*

Nakajima, S., & Allen, J. 1993. A Study on Prosody and Discourse Structure in Cooperative Dialogues. *Pages 197–210 of: Phonetica*, vol. 50.

Ney, H., Essen, U., & Kneser, R. 1994. On structuring probabilistic dependencies in stochastic language modeling. *Computer Speech and Language*, **8**(1), 1–28.

O'Connor, J. D., & Arnold, G. F. 1973. *Intonation of Colloquial English.* 2 edn. Longman.

Palmer, H. 1922. *English Intonation with Systematic Exercises.* Cambridge Universtity Press.

Pierrehumbert, J. B. 1980. *The Phonology and Phonetics of English Intonation.* Ph.D. thesis, MIT. Published by Indiana University Linguistics Club.

Pierrehumbert, J. B., & Hirschberg, J. 1990. The meaning of intonational contours in the interpretation of discourse. *In:* **Cohen, P. R., morgan, J., & Pollack, M. E.** (eds), *Intentions in Communication.* MIT press.

Pitrelli, J. F., Beckman, M. E., & Hirschberg, J. 1994. Evaluation of prosodic transcription labeling reliability in the ToBI framework. *Pages 123–126 of: Proceedins of ICSLP '94*, vol. 1.

Poesio, M., & Mikheev, A. 1998. The Predictive Power of Game Structure in Dialogue Act Recognition: Experimental Results Using Maximum Entropy Estimation. *In: Proceedings of ICSLP '98.*

Power, R. 1979. The organization of purposeful dialogues. *Linguistics*, **17**, 107–152.

Rabiner, L., & Juang, B.-H. 1993. *Fundamentals of speech signal processing.* Prentice Hall.

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

Reithinger, N., Engel, R., Kipp, M., & Klesen, M. 1996. Predicting Dialogue Acts for a Speech-to-Speech Translation System. *Pages 654–657 of: Proceedings of ICSLP '96.*

Reithinger, N., , & Klesen, M. 1997. Dialogue Act Classification using Language Models. *Pages 2235–2238 of: Proceedings of Eurospeech '97.*

Rosenblatt, F. 1958. The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Pages 386–408 of: Psycological Review*, vol. 65.

Rosenfeld, R., & Clarkson, P. 1997. *CMU-Cambridge Statistical Language Modeling Toolkit v2.* `http://svr-www.eng.cam.ac.uk/~prc14/`.

Ross, K., & Ostendorf, M. 1995. A dynamical system model for recognising intonation patterns. *Pages 993–996 of: Proceedings of Eurospeech '95.*

Rumelhart, D., & McClelland, J. 1986. *Parallel Distributed Processing.* MIT Press, Cambridge, MA.

Sag, I., & Liberman, M. Y. 1975. The intonational disambiguation of indirect speech acts. *Pages 487–497 of: Proceedings of the Chicago Linguistics Society,* vol. 11.

Schegloff, E., & Sacks, H. 1973. Opening-up closings. *Pages 289–327 of: Semiotica.*

Searle, J. 1975. *Language, Mind and Knowledge. Minesota Studies in the Philosphy of Science.* University of Minesota Press. Chap. A Taxonomy of Illocutionary Acts.

Sejnowski, T., & Rosenberg, C. 1986. *NETtalk: a Parallel Network that Learns to Read Aloud.* Tech. rept. JHU/EECS-86/01. The John Hopkins University of Electrical Engineering and Computer Science Technical Report.

Shriberg, E., Bates, R., & Stolcke, A. 1997. A prosody-only decision-tree model for disfluency detection. *In: Proceedings of Eurospeech '97.*

Shriberg, E., Taylor, P., Bates, R., Stolcke, A., Ries, K., Jurafsky, D., Coccaro, N., Martin, R., Meteer, M., & Ess-Dykema, C. V. 1998. Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech? *Language and Speech,* **41**(3-4), 439–487.

Siegel, S., & Castellan, N. 1988. *Nonparametric Statistics for the Behavioral Sciences.* 2nd edn. McGraw-Hill.

Silverman, K. 1988. Utterance-internal Prosodic Boundaries. *Pages 86–91 of: Proceedings of the Second Australian Internation Conference on Speech Science Technology.* Australian Speech Science and Technology Association, Sydney.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., & Hirschberg, J. 1992. ToBI: a standard for labelling English prosody. *Pages 867–870 of: Proceedings of ICSLP '92*, vol. 2.

Silverman, K., & Pierrhumbert, J. 1990. Papers in laboratory phonology. vol. 1. Cambridge University Press.

Sinclair, J. M., & Coulthard, M. 1975. *Towards an analysis of discourse : the English used by teachers and pupil.* Oxford University Press, London.

SNNS. 1997. *Stuttgart Neural Network Simulator.* `http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html`.

Stolcke, A., & Shriberg, E. 1996. Automatic linguistic segmentaion of conversational speech. *Pages 1005–1008 of: Proceedings of ICSLP '96*, vol. 3.

Stolcke, A., Shriberg, E., Bates, R., Taylor, P. A., Ries, K., Jurafsky, D., Coccaro, N., Martin, R., Meteer, M., Ries, K., , & Ess-Dykema, C. V. 1998. Dialog act modeling for conversational speech. *Pages 98–105 of: Applying Machine Learning to Discoure Processing.*

Taylor, P. A., King, S., Isard, S. D., Wright, H., & Kowtko, J. 1997. Using Intonation to Constrain Language Models in Speech Recognition. *In: Proceedings of Eurospeech '97.*

Taylor, P. A. 1995. Using Neural Networks to Locate Pitch Accents. *In: Proceedings of Eurospeech '95, Madrid.*

Taylor, P. A. 1998. The Tilt Intonation Model. *In: Proceedings of ICSLP '98.*

Taylor, P. A. 2000. Analysis and Synthesis of Intonation using the Tilt Model. *Journal of the Acoustical Society of America*, **107**.

Taylor, P. A., King, S., & Black, A. 1998a. *Edinburgh Speech Tools.* Available from the Centre for Speech Technology Research `http://www.cstr.ed.ac.uk/projects/speech_tools/manual/speechtools_toc.html`. Email {`pault,simonk,awb`}`@cstr.ed.ac.uk`.

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

Terry, M., Sparks, R., & Obenchain, P. 1994. Automated Query Identification in English Dialogue. *In: Proceedings of ICSLP '94.*

t'Hart, J., & Cohen, A. 1973. Intonation by rule: a perceptual quest. *Journal of Phonetics*, **1**, 309–327.

t'Hart, J., Collier, R., & Cohen, A. 1990. *A perceptual study of intonation: an experimantal-phonetic approach.* Cambridge University Press.

Traum, D., & Hinkelman, E. 1992. Conversation Acts in Task-Oriented Spoken Dialogue. *Computational Intelligence*, **8**(3), 575–599.

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically imal decoding algorithm. *IEEE Trans., Information Theory*, **13**, 260–269.

Wahlster, W. 1993. Verbmobil - Translation of Face-To-Face Dialogs. *Pages 29–38 of: Proceedings of Eurospeech '93.*

Warnke, V., Kompe, R., Niemann, H., & Noth, E. 1997 (Sept.). Integrated dialogue act segmentation and classification using prosodic features

and language models. *Pages 207–210 of: Proceedings of Eurospeech '97*, vol. 1.

Wightman, C., & Ostendorf, M. 1994. Automatic Labelling of Prosodic Patterns. *Pages 469–481 of: IEEE Trans. on Speech and Audio Proc*, vol. 2.

Witten, I. H., & Bell, T. C. 1991. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Trans. on Information Theory*, **37**(4).

Woszczyna, M., & Waibel, A. 1994. Inferring Linguistic Structure in Spoken Language. *In: Proceedings of ICSLP '94.*

Wright, H. 1998. Automatic Utterance Type Detection Using Suprasegmental Features. *In: Proceedins of ICSLP '98.*

Wright, H., & Taylor, P. A. 1997. Modelling Intonational Structure using Hidden Markov Models. *In: ESCA workshop on Intonation: Theory Models and Applications.*

Wright, H., Poesio, M., & Isard, S. I. 1999. Using High Level Dialogue Information for Dialogue Act Recognition using Prosodic Features. *In: ESCA Workshop on Dialogue and Prosody.*

Young, S., Jansen, J., Odell, J., Ollason, D., & Woodland, P. 1996. *HTK manual.* Entropic.