Intonation and dialogue context as constraints for speech recognition

Paul Taylor
Simon King
Stephen Isard
Helen Wright

Centre for Speech Technology Research, University of Edinburgh,
80, South Bridge, Edinburgh, U.K. EH1 1HN
http://www.cstr.ed.ac.uk
email: {pault, simonk, stepheni, helen}@cstr.ed.ac.uk

Acknowledgements
This work benefitted from many conversations with our colleagues Jacqueline Kowtko and Hiroshi Shimodaira.
We are pleased to acknowledge the support of the UK Engineering and Physical Science Research Council through EPSRC grant GR/J55106. Simon King's work was funded by Reuters. Helen Wright holds an EPSRC studentship.

Running head:
Intonation and dialogue context as constraints for speech recognition

Abstract

This paper describes a way of using intonation and dialogue context to improve the performance of an automatic speech recognition (ASR) system. Our experiments were run on the DCIEM Maptask corpus, a corpus of spontaneous task-oriented dialogue speech. This corpus has been tagged according to a dialogue analysis scheme that assigns each utterance to one of 12 “move types”, such as “acknowledge”, “query-yes/no” or “instruct”. Most asr systems use a bigram language model to constrain the possible sequences of words that might be recognised. Here we use a separate bigram language model for each move type. We show that when the “correct” move-specific language model is used for each utterance in the test set, the word error rate of the recogniser drops. Of course when the recogniser is run on previously unseen data, it cannot know in advance what move type the speaker has just produced. To determine the move type we use an intonation model combined with a dialogue model that puts constraints on possible sequences of move types, as well as the speech recogniser likelihoods for the different move-specific models. In the full recognition system, the combination of automatic move type recognition with the move specific language models reduces the overall word error rate by a small but significant amount when compared with a baseline system that does not take intonation or dialogue acts into account. Interestingly, the word error improvement is restricted to “initiating” move types, where word recognition is important. In “response” move types, where the important information is conveyed by the move type itself - e.g., positive vs. negative response - there is no word error improvement, but recognition of the response types themselves is good. The paper discusses the intonation model, the language models and the dialogue model in detail and describes the architecture in which they are combined.

INTRODUCTION

This paper describes a strategy for using a combination of intonation and dialogue context to reduce word error rate in a speech recognition system for spontaneous dialogue. Although databases of conversational speech have been taken up as a major challenge by the speech recognition community in recent years, the architecture of recognition systems, originally developed for read speech and/or isolated utterances, has not been adapted to take into account the ways in which conversational speech is different. At the same time, systems intended for computer-human dialogue have tended to adopt existing speech recognisers as black box front ends, rather than using dialogue information to guide recognition. In contrast, the work we report here builds properties of conversational speech into the architecture of the recogniser.

Central to our approach is the concept of *dialogue acts*, such as queries, responses and acknowledgements. Our system exploits three properties of these kinds of acts: they have characteristic intonation patterns, they have characteristic syntax and they tend to follow one another in characteristic ways.

For each category of act we have a separate intonation model, reflecting, for instance, that genuine information-seeking yes/no questions tend to rise in pitch at the end, while acknowledgments of instructions tend to have low pitch without prominent accentuation. Applying each of these models to the F_0 and energy contours of an utterance gives us a set of *intonational likelihoods* for the utterance being one or another type of dialogue act.

At the same time, we have a separate language model for each type of dialogue act, to take into account, for instance, the greater probability of a yes/no query beginning with an auxiliary inversion (“is there...”, “do you...”) while an acknowledgement is likely to contain an affirmative word like “okay”. Running the speech recogniser with each of the different language models gives us a set of *language model likelihoods* for the utterance’s dialogue act type.

Finally, we have a *dialogue model* that assigns probabilities to sequences of dialogue acts coming one after another. For instance, a query followed by a response followed by an acknowledgement is more likely than three acknowledgements in succession.

Our system proceeds by combining the likelihoods from all three models to find the most likely dialogue act sequence for a series of utterances, and then it adopts the recognition results from the language models corresponding to that sequence of dialogue acts. For example, the high likelihood of one utterance being a yes/no query might strengthen the case for the following one being a reply to it, and so support the interpretation of an indistinct word at the beginning as “yeah”.

There are a number of different dialogue act schemes currently used by computer dialogue systems (e.g., (Lewin *et al.*, 1993), (Reithinger *et al.*, 1996), (Allen *et al.*, 1995)). There is also an initiative to develop a standard scheme (Carletta *et al.*, 1997a). The work we report here was done on the DCIEM

Maptask corpus (Bard *et al.*, 1995), a corpus of spontaneous task-oriented dialogue speech collected from Canadian speakers of English, and our dialogue analysis is based on the theory of *conversational games* first introduced by Power (1979) and adapted for Maptask dialogues as described in (Carletta *et al.*, 1997b). In this system individual dialogue acts are referred to as *moves* in the conversational games.

While the existing dialogue act schemes differ at various points, they are broadly similar, and the methods described in this paper should be straightforwardly transferrable to any of the others. It is worth noting that identification of dialogue acts, which we treat here as just a means to the end of better word recognition, becomes an end in itself in dialogue systems such as those mentioned above that actually engage in dialogues with human users.

THE DCIEM DIALOGUES

The experiments here use a subset of the DCIEM Maptask corpus (Bard *et al.*, 1995)[1]. The two participants in a dialogue have different roles referred to as *(instruction) giver* and *(instruction) follower*. It is the giver's task to guide the follower along a route on the map. Because of their different roles, the giver and follower have different distributions of moves.

The data files were transcribed at the word level and divided into utterances, each corresponding to a single move. The speech files were also hand labelled with the intonation scheme described in the Intonational Events section below. Forty five dialogues (9272 utterances) were used for training the recognition system, and five dialogues (1061 utterances) were used for testing it. None of the test set speakers were in the training set, so the results we report are speaker independent. The language models and the HMM phone models were all trained on the full set of forty five dialogues. The intonation model was trained on a hand labelled subset of twenty dialogues.

Conversational Game Analysis

The conversational game analysis described in (Carletta *et al.*, 1997b) uses six games: *Instructing, Checking, Query-YN, Query-W, Explaining* and *Aligning*. The initiating moves of these games are described in Table 1, and other possible moves in Table 2. We find that our use of intonation and dialogue context improves word recognition accuracy for the initiating moves, but not for the rest. On the other hand, the initiating moves contain a relatively higher

[1] The DCIEM corpus of Canadian speech was chosen in preference to the original Glasgow Maptask corpus (Anderson *et al.*, 1991) because it allowed us to exploit the large body of previous work on North American speech to build a better baseline speech recogniser than we could achieve by starting from scratch with Glasgow speech. The DCIEM corpus contains a number of dialogues recorded in sleep deprived and other non-standard conditions, but none of these were included in our subset.

proportion of content words, which need to be recognised correctly, while the important information in non-initiating moves is often conveyed by the move type itself; mistaking “yep” for “yeah” in a *Reply-y* is not likely to derail a dialogue.

Instruct	direct, or indirect request or instruction. E.g. “Go round, ehm horizontally underneath diamond mind...”
Explain	provides information, believed to be unknown by the game initiator. E.g. “I don’t have a ravine.”
Align	checks that the listener’s understanding aligns with that of the speaker. E.g. “Okay?”
Check	asks a question to which the speaker believes s/he already knows the answer, but isn’t absolutely certain. E.g. “So going down to Indian Country?”
Query-yn	a yes-no question. E.g. ”Have you got the graveyard written down?”
Query-w	asks a question containing a wh-word. E.g. “In where?”

Table 1: Initiating moves

SYSTEM ARCHITECTURE

Our technique is based on the idea that by giving a speech recogniser different language models[2] for recognising different move types, we can achieve better recognition results. Such an approach is not likely to be successful unless:

1. Most of the individual language models describe their own move types more accurately than a general language model does
2. We have a way of invoking the right model at the right time, to take advantage of 1.

[2] We adopt standard speech recognition terminology and use the term *language model* for a device that assigns probabilities of occurrence to strings of words. This contrasts with *dialogue models* that assign probabilities to sequences of dialogue moves, without regard to the specific words that constitute them.

Acknowledge	indicates acknowledgement of hearing or understanding. E.g.“Okay.”
Clarify	clarifies or rephrases old information. E.g. { so you want to go ... actually diagonally so you’re underneath the great rock.} “diagonally down to uh horizontally underneath the great rock.”
Reply-y	elicited response to query-yn, check or align, usually indicating agreement. E.g. “Okay.”, “I do.”.
Reply-n	elicited response to query-yn, check or align, usually indicating disagreement. E.g. “No, I don’t.”.
Reply-w	elicited response that is not to clarify, reply-y or reply-n. It can provide new information and is not easily categorizeable as positive or negative. E.g. { And across to?} “The pyramid.”.
Ready	indicates that the previous game has just been completed and a new game is about to begin. E.g. “Okay.”, “Right,” {so we’re down past the diamond mine?}

Table 2: Other moves

The first of these two items is dealt with in the section on language modelling below. Other systems such as (Eckert *et al.*, 1996; Baggia *et al.*, 1997) have made similar use of dialogue state dependent language models to improve recognition. It is in addressing the second item that our approach differs, in that our choice of which language model to use is integrated into the recognition process, rather than being based simply on the system’s record of the state of the dialogue. Our choice of language model is arrived at by combining the move type likelihoods provided by our dialogue model, our intonation models, and, in effect, several copies of the speech recogniser, each of which uses a different language model.

Figure 1 illustrates the process of determining the optimal sequence of move types. For each possible sequence of move types, we combine their intonational and speech recognition likelihoods, as well as the dialogue model likelihood for the sequence itself. Although conceptually one can imagine this being done by exhaustive enumeration of the possible move sequences, we adopt the computationally efficient alternative of Viterbi search. The mathematical formulation is presented in the appendix at the end of the paper. The relative contributions of the intonational and speech recognition likelihoods are weighted using factors that are optimised on the training data.

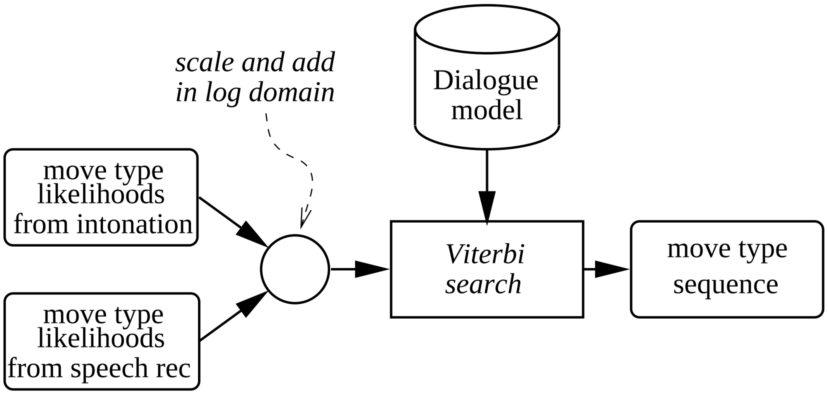
scale and add
in log domain
Dialogue model
move type likelihoods from intonation
move type likelihoods from speech rec
Viterbi search
move type sequence

Figure 1: Finding the best move sequence

The bottom level speech recogniser that provides the word hypotheses that the language models constrain is an HMM based system built using the HTK toolkit (Young *et al.*, 1996) in a standard configuration[3]. Approximately three hours and twenty minutes of speech was used to train the models. Using a single language model derived from the entire training set, the recogniser achieves a word error rate of 24.8%. This is the baseline result that we are trying to improve on by introducing separate move-specific language models in the way just described.

DIALOGUE MODELLING

For purposes of predicting the identity of the next move from dialogue context, we use a very simple sort of dialogue model which gives probabilities based on

1. current speaker role (giver or follower)
2. move type of *other* speaker's most recent move
3. role of speaker of immediately preceding move

where 2 and 3 may refer to the same move (when the speakers take alternating moves).

We arrived at this model by examining various N-gram (Jelinek & Mercer, 1980) types using different sets of predictors and choosing the one that gave the

[3] 12 cepstral co-efficients plus energy, plus their first and second derivatives, giving 39 component observation vectors, and 8-component Gaussian mixture tied-state cross-word triphone models. See, e.g., (Young *et al.*, 1996; Rabiner & Juang, 1994)

best predictive power (i.e., lowest perplexity, see below) on a held out portion of the training set. Our chosen model, which uses three items to predict a fourth, is classified as a 4-gram model.

The dialogue model was trained on the same data set as the language models for speech recognition. At run time, we assume speaker roles – items 1 and 3 above – are known, but item 2 is the automatically recognised move type.

Table 3 compares the perplexity of our 4-gram dialogue model with simple unigram and bigram models. The unigram model simply reflects the relative frequency of the various move types, regardless of context, and the bigram model uses the preceding move type, regardless of speaker, to predict the current move type. The models were trained on the entire training set, but tested on the *test* set. These figures are therefore for illustration only and were not used in the choice of dialogue model.

Model	**Test set perplexity**
unigram	9.1
bigram	6.3
4-gram	5.2

Table 3: Dialogue model perplexities (12 move types)

Intuitively, perplexity can be thought of as a measure of how much more information is needed to correctly classify some item as belonging to one of a number of classes. As an information theoretic measure, it has nothing to say about the content of the information needed, just the quantity. If there are N classes that the item might be assigned to, all equally likely, then the perplexity is N. If some of the classes are more likely than others, then the perplexity works out to less than N, which means that the amount of information required is the same as for some smaller number of equiprobable classes. (The limiting case where one class is certain and all others impossible corresponds to a perplexity of 1 - there is just one possibility.)

What Table 3 then tells us is that taking into account the unequal frequencies of the twelve different move types makes predicting the next move about as hard as with nine equiprobable types, but taking the contextual probabilities given by the bigram or 4-gram models into account reduces the difficulty to slightly more than predicting with six or five equiprobable classes, respectively.

IDENTIFYING MOVES BY INTONATION

In order to integrate intonational analysis into the probabalistic framework required for speech recognition, we have adopted a novel approach to intonational phonology and its relationship to conversational structure. Accounts of intonational meaning normally attribute a discourse function either to whole tunes like O'Connor and Arnold's (1973) *high drop* or Sag and Liberman's (1975) *surprise/redundancy* tune, or to particular types of accent, possibly with rules for composing meanings when accents appear in combination, as in (Pierrehumbert & Hirschberg, 1990). Such accounts are often insightful, but, being pitched at the phonological level, they are concerned with idealised cases of pitch contours. A recognition system has to cope with contours that are not clearly classifiable as one tune type or another, and with the possibility that an apparently clear case of a tune or accent type is associated with the "wrong" meaning. In the phonetic domain, Markov models have been successfully employed to represent the range of variation among spectra associated with a given phoneme. Here we use Markov models in a similar way to represent the range of contours associated with a given dialogue act.

It is already common practice in intonational phonology to present possible sequences of basic intonational elements, such as pitch accents or boundary tones, by way of a finite state network. Figure 2a shows the familiar form of Pierrehumbert's intonational grammar giving her account of the legal tone sequences of English (Pierrehumbert, 1980). For present purposes, it is useful to rewrite this grammar in a form where symbols are emitted from states rather than from arcs. Figure 2b shows the Pierrehumbert grammar in this alternative form in which the pitch accent state emits all the pitch accent types, and the self-transition arc shows that this state can be visited multiple times. Figure 2c shows Ladd's (1996) amended version where nuclear accents are treated differently from pre-nuclear accents. Figure 2d shows the British School system of pre-head, head, nucleus and tail.

Such networks can be turned into Markov models by adding two types of probabilities. *Transition probabilities* are associated with arcs between states which give, for example, the likelihood of a contour having or not having a prehead. *Observation probabilities* are associated with states and give the relative frequencies of the types that the state can emit. For example, the pitch accent state in figure 2b might have a high chance of emitting a common accent such as H* and a much lower chance of emitting a rarer accent such as H+L*.

In our training data, each move type has a distribution of intonational event (observation) sequences associated with it, and we model each of these distributions with a separate Markov model. We use a model with three states, and include self-transition arcs to all states, making it possible for them to repeat. Given the type of intonational observations we use (described below), any observation can potentially be generated by any state, even though some observations are more probable from some states than from others. It is therefore not possible to say with complete certainty which state a given observation is associated with. A Markov model with this property is commonly referred to as a *hidden Markov model* (HMM) because the state sequence is not deterministically recoverable from the observation sequence.

Hidden Markov models can be trained using the Baum-Welch algorithm (Baum, 1972) to provide optimal transition and observation probabilities for

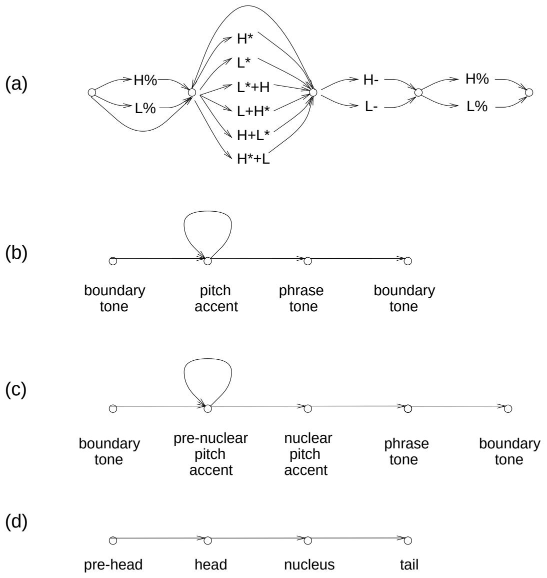

Figure 2: Intonational structure represented by finite state networks

modelling their particular training data. As long as each move type has a different distribution of observations in the training data, its hidden Markov model will have different transition and observations probabilities from those of the other moves.

When confronted with the sequence of intonational events from a previously unseen utterance, we can calculate the probability that each of our models might have produced it. These probabilities are taken as the intonational contribution to the identification of the utterance's move type.

Intonational Events

The Markov model framework just described puts no constraints on the form that intonational events can take. The schemes depicted in figure 2 each have a finite repertoire of discrete categories of events. For instance, in the ToBI system (Silverman *et al.*, 1992), derived from Pierrehumbert's work, there are five pitch accents, two phrase accents and two boundary tones. We have chosen instead to use just a single category of *event*, but our events are characterised by real number parameters, rather than being a discrete set.

We have avoided discrete intonational categories for several reasons. First, even on clear read speech human labellers find it notoriously difficult to label the categories reliably, and the reliability drops further for spontaneous speech. In a study on ToBI labelling (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% and this figure was only achieved by first collapsing some of the main categories (e.g. H* with L+H*). Second, the distribution of pitch accent types is often extremely uneven. In a portion of the Boston Radio news corpus which has been labelled with ToBI, 79% of the accents are of type H*, 15% are L*+H and other classes are spread over the remaining 6%. From an information theoretic point of view, such a classification isn't very useful because virtually everything belongs to one class, and therefore very little information is given by accent identity. Furthermore, not all H* accents have the same linguistic function, and so there are intonational distinctions that are missed by only using a single broad category. Finally, recognition systems which have attempted to automatically label intonation usually do much better at the accent detection task than at classifying the accents (e.g. (Ross & Ostendorf, 1995)).

In brief then, we choose a single category of accent, because both human and automatic labellers find it difficult to distinguish more, and because even if it were possible to distinguish them the payoff in information would be small. To put it another way, in practical situations the ToBI system more or less equates to a single pitch accent type anyway - all we have done is to make this explicit.

However, this is not to say that we believe that all pitch accents are identical, just that current categorical classification systems aren't suited for our purposes. To classify pitch accents, we use four continuous parameters collectively known as *tilt* parameters.
The tilt parameters are:

- F_0 at the start of the event
- The *amplitude* of the event
- The *duration* of the event
- *Tilt*, a measure of the shape of the event

The tilt parameters are derived from automatic analysis of the shape of the F0 contour of the event. The first stage in this process is known as RFC (rise/fall/connection) analysis (Taylor, 1995). In the RFC model, each event can consist of a rise, a fall, or a rise followed by a fall. RFC analysis begins by locating rises and/or falls in a smoothed version of the event's F0 contour. Piecewise quadratic curves are fitted to each rise or fall, and the start and end points of these curves are marked, from which the rise amplitude, the fall amplitude, the rise duration and the fall duration can be calculated, as illustrated in figure 3. When the event consists of only a rise or only a fall, the amplitude and duration for the missing part are set to 0. The RFC parameters are then converted to tilt parameters which are more amenable to linguistic interpretation.

Tilt is meant to capture the relative amounts of rise and fall in an event. It can be measured from the rise and fall amplitudes:

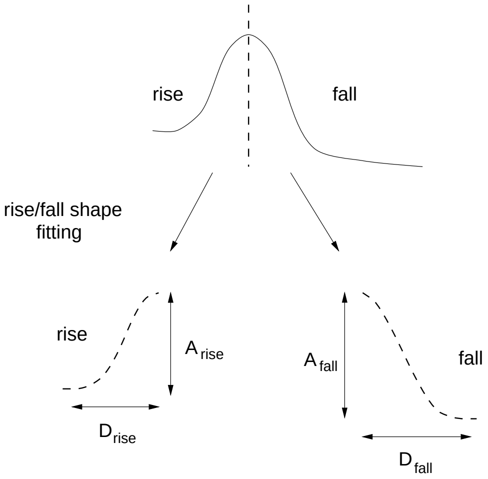

Figure 3: $A_{rise}, D_{rise}, A_{fall}$ and D_{fall} are derived from separate curves fitted to the rise and fall portions of an event.

$$tilt_{amp} = \frac{|A_{rise}| - |A_{fall}|}{|A_{rise}| + |A_{fall}|} \tag{1}$$

or the rise and fall durations:

$$tilt_{dur} = \frac{D_{rise} - D_{fall}}{D_{rise} + D_{fall}} \tag{2}$$

Experimental studies (Taylor, 1998) have shown that these two quantities are highly correlated, and hence with little loss of information they can be combined into a single quantity, taken as the average of the two:

$$tilt = \frac{A_{rise} - A_{fall}}{2(A_{rise} + A_{fall})} + \frac{D_{rise} - D_{fall}}{2(D_{rise} + D_{fall})} \tag{3}$$

Figure 4 shows the tilt values for several different contour shapes.
The amplitude and duration are calculated from the combined amplitudes and durations of the rise and fall components.

$$A_{event} = |A_{rise}| + |A_{fall}| \tag{4}$$

$$D_{event} = |D_{rise}| + |D_{fall}| \tag{5}$$

Event Detection Event detection is also performed by HMMs, using as observations F_0 and rms energy at 10ms intervals, together with standard rate of change and acceleration measures ("deltas"). The means and variances for each speaker's F_0 and energy are calculated and used to normalise the data for that speaker.

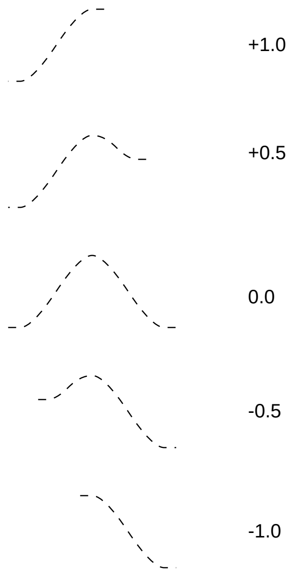

Figure 4: Variation of contour shape as a function of the tilt parameter.

A continuous density HMM with 8 Gaussian components is trained for each of 5 labels: **a** is used for pitch accents, **b** for boundary tones and a compound label **ab** is used for the case when an accent and boundary are so close that they overlap and form a single intonational event. **sil** is used for silence and **c** is used for the parts of the contour which are not an event or silence. The HMMs are trained on 20 dialogues from the training set which have been hand labelled with these labels. The standard Baum-Welch algorithm is used for training.

Once trained, the system is run by using the HMMs for each label in combination with a bigram model representing the prior probabilities of pairs of labels occurring in sequence. The Viterbi decoding algorithm is used to determine the most likely sequence of labels from the acoustics of the utterance being recognised.

The distinction among **a**, **b** and **ab** is dropped when tilt parameters are calculated. The use of three separate event labels is to some extent historical, since they were present in our hand labelled database, but we also found that that the system performs better at distinguishing events from non-events using the three categories, even though it is not particularly accurate in distinguishing among them.

Event Detection and Move Identification Results

We report here performance results for intonational event detection and the conversational move identification, in isolation from the rest of the system.

Test	**% Correct**
Unigram on all moves	42
Unigram on initiating moves	36
Unigram on other moves	48
4-gram on all moves	47
4-gram on initiating moves	41
4-gram on other moves	52

Table 4: Results for move identification

Intonation Event Detector Event detection performance can be measured by comparing the output of the recogniser with the hand labelled test set. We are only concerned with placement of event labels (**a**, **b** and **ab**), since it is just the sequence of events that acts as input to the move recogniser. An automatically labelled event is counted as correct if it overlaps a hand labelled event by at least 50%. Using this metric 74.3% of hand labelled events are correctly identified. However, the other standard measure of recogniser performance, namely *accuracy*, calculated as

$$\frac{correctly\ recognised\ events - incorrect\ events\ inserted\ by\ the\ recogniser}{total\ number\ of\ hand{-}labelled\ events}$$

is 47.7%

These results are not as bad as they might at first appear. First of all, when the speech was labelled, the labellers were allowed to use a diacritic “minor” to mark accents. This was used either when accents were very small, or when the labellers were unsure of the presence of the accent at all. If accents with this diacritic are ignored, 86.5% of the remaining accents are correctly identified, so nearly half the missed accents are of this marginal sort.

The difference between percent correct and accuracy means that the recogniser inserts a lot of spurious events. These spurious events are in general of low amplitude. The move recogniser is trained on the output of the event recogniser and so as long as the event recogniser produces the same pattern of small amplitude spurious events on test sentences as it does on training sentences, they are at worst just a source of noise. If the spurious events are in fact more likely to occur in one type of move than another, then the move recogniser will learn to exploit them. If the spurious events are not correlated with move type, which we believe to be the case, then a move recogniser trained on events produced by the event recogniser will assign a higher probability to small amplitude events in all move types than a move recogniser trained on hand labelled data. However, it will not depend on these minor events to distinguish one move type from another.

Move Identification Table 4 gives a summary of results for identification of the 12 move types, using the output of the intonational move recogniser in conjunction with two different sorts of dialogue model. A unigram is the simplest type of dialogue model and just gives the prior probability of each move occurring. The overall performance of the move recogniser with this type of dialogue model is 42%. This result improves when the 4-gram described earlier is used. Furthermore, we can see that non-initiating moves are better identified than initiating moves in both cases.

We actually tried two variants of the 4-gram language model: the *overhearer* and the *participant* scenarios. For the latter, the computer is imagined as participating in the task and is assumed to know the identity of its own most recent move[4]. This makes the task a bit easier, but in general we have found that results from participant mode and overhearer mode are surprisingly similar, so we report only the results for the slightly harder overhearer version of the recognition task.

LANGUAGE MODELLING

As explained in the System Architecture section above, a necessary, though not sufficient, condition for our approach to succeed is that move specific language models (LMs) should assign higher probability than a general LM to utterances of “their” move type, in order to encourage better recogniser performance on utterances of that type. High average probability on a move set equates to low perplexity. In this section we give relative perplexity results for a general LM and several variants of move specific LMs.

Training set

The training set was divided into move specific sections. The total number of training tokens (words) per move type is given in table 5. If we compare the two rightmost columns, we can see that the average sentence length varies widely across move types, so although some types are relatively infrequent (fewer than 3% of moves are *clarify*, for example), there are still sufficient training tokens for these types. The exceptions to this pattern, such as *reply-n*, tend to have simple grammars anyway, so training data is not as sparse as suggested by the number of tokens in the training set. Given the amount of training data available, bigrams were the only practical choice for N-gram language models.

[4] A reviewer correctly points out that the partner’s next move will be based on *the partner’s interpretation* of what the speaker has just said, which does not necessarily coincide with the move that the speaker intended to make. However, the assignment of move types to utterances in our data is based on the transcriber’s interpretation of the speaker’s intention. The fact that the response is not always appropriate to that intention is reflected in the dialogue model, where non-zero probabilities are sometimes assigned to “impossible” utterance sequences.

Move type	Utterances	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 5: Move type specific LM training set sizes

Language model smoothing

To compensate for sparcity of training data, two techniques were used: backing-off and smoothing. Both the "general purpose" and move specific bigram LMs were backed off language models (Church & Gale, 1991). Smoothing of the grammars was achieved by taking weighted averages of the move specific bigram probabilities and the corresponding bigram probability from the general purpose LM. The weights were chosen by a maximum likelihood method, using a held-out scheme (that is, by dividing the training set itself into training and testing portions) with the CMU Language Modelling toolkit (Rosenfeld & Clarkson, 1997). As expected, the weights for different move types varied widely. We would expect the smoothed versions of move specific LMs which are well-estimated, and which are markedly different from the general purpose LM, to consist mainly of the move specific LM, and be less dependent on the general purpose LM. This proves to be the case; for example, the smoothed LM for *acknowledge* consists of 0.8 acknowledge LM and 0.2 general purpose LM, while for *clarify* these weights are 0.3 and 0.7 respectively.

Perplexity results

The choice of language model was based on perplexity on a held-out portion of the training set. Here we give perplexity results for the *test* set for consistency with other results.

In table 6, we see that the perplexities vary widely between move types and that sometimes the move-specific language model perplexities are much higher (worse) than those for the general model. This is the case for *align*, *clarify* and *reply-w* in particular. We presume this is because of insufficient

Move type	**Perplexity**		
	Language model used		
	general	move specific	smoothed
acknowledge	4·3	3·5	3·4
align	22·1	31·7	22·1
check	32·4	35·5	32·3
clarify	46·8	60·6	46·8
explain	40·7	42·4	41·3
instruct	41·4	37·2	37·7
query-w	36·6	34·6	32·3
query-yn	20·5	19·3	19·0
ready	4·0	2·6	3·0
reply-n	7·5	3·0	3·8
reply-w	24·0	32·3	25·0
reply-y	7·0	4·6	5·1

Table 6: Perplexity of general and move-specific models, by move type

training data for these types.

Furthermore, the smoothed move specific models do not always have a lower perplexity than the unsmoothed ones because the smoothing weights are not estimated on the test set. By computing the perplexity of all models on held out training data (not the same data used to compute the smoothing weights in the first place), we can estimate whether the smoothed, unsmoothed or general purpose model will be best (on test data) for each move type. We then choose the model with the lowest estimated perplexity for each move type – we call this the *best choice* model. Table 7 compares the overall perplexities of the unsmoothed, smoothed and best choice models. In this case, the general purpose model was taken as best choice for move types *clarify*, *explain* and *reply-w*. The figures in Table 6 show that this is a good decision based on the test set.

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

Table 7: Language model perplexities

SYSTEM PERFORMANCE RESULTS

As explained earlier, the speech recogniser is run with each language model over each utterance. The language model likelihoods produced are combined with the intonation likelihoods and the dialogue model to provide a single best sequence of moves for the whole conversation. The final recognition output for a given utterance is then the word string that was recognised using the language model of the move type chosen for that utterance.

Tables 8 gives results for word error rate (calculated as $100\% - accuracy$) in several recognition experiments. The baseline figures are obtained by running the speech recogniser using a single general purpose language model, with no reference to move types, dialogue or intonation. The "cheating" figures give the performance of our system using the correct move-specific language model every time, corresponding to 100% accuracy of the move classifier. They represent the best result we could possibly have achieved with our techniques on this set of test data, using our current move classification scheme and its associated bigram language models. These figures are better (lower error rate) than the corresponding baseline ones, showing that the perplexity figures of tables 6 and 7 translate to an improvement in recognition performance. The reductions in error rate for all utterances and for initiating moves taken separately are significant by a paired t-test ($p < 0.0005$).

The scores for automatic move recognition fall between those for the baseline and for perfect recognition, though they are closer to the latter. Going through the motions of a paired t-test to compare the overall recognition score with the baseline would appear to produce a significant result, but the test is not strictly applicable in this case, because use of the 4-gram dialogue model means that recognition is not independent for successive utterances, violating the assumptions of the test. However, given the nature of the 4-gram, it is probably safe to treat initiating moves as independent each from the next, and similarly for non-initiating moves. On this basis, the 1.3% difference (5% reduction) in error rate between the baseline and the full system on initiating moves is significant ($p < 0.001$). The slight deterioration for non-initiating moves is not significant.

Given that the role of intonation and dialogue context in our system is to help find the right language model for recognising each utterance, it is worth considering whether the more straightforward strategy of simply choosing the language model giving the best recognition score would work as well. The bottom section of the table shows the performance of this alternative strategy. For this case, we just chose the result from the move specific language model that "was the most confident", i.e., assigned the highest likelihood to its output, ignoring intonation and context. Here the improvement for initiating moves is significant ($p < 0.005$), as is the deterioration for non-initiating moves ($p < 0.05$), but not the overall improvement. (Since the dialogue model is not involved in this case, the independence assumption of the paired t-test is satisfied.)

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

Table 8: System performance compared with baseline

We have also examined percentage agreement on move type between the system as a whole and various components taken on their own. Intonation and dialogue model alone agree with the whole system 78% of the time, while the language model likelihoods alone agree only 47% of the time. Intonation and dialogue alone correctly identify the move type 47% of the time, as shown in table 4. The system as a whole is correct 64% of the time, which subdivides into 54% for initiating moves and 80% for non-initiating. In particular, there is only one confusion in the entire test set between *Reply-y* and *Reply-n* and that is for a case where the transcribers labelled a “no” answer to a “you don’t ..., do you?” question as a *Reply-y*, but the system called it a *Reply-n*. Language model likelihoods alone correctly identify only 40% of moves.

DISCUSSION

The reduction in error rate that we achieve is roughly comparable to that reported by others (Eckert *et al.*, 1996; Baggia *et al.*, 1997) who have employed dialogue context dependent language models. Detailed comparisons are not possible because of domain and task differences. The main limitation on our results is the relatively small gap between baseline performance and the best performance achievable with perfect move recognition. Possible ways of widening the gap include an improved dialogue act set, more sophisticated kinds of language models and, of course, as always in speech recognition, more training data. Once the gap has been widened, there is scope for improved intonation recognition, possibly using the CART classification techniques discussed in (Shriberg *et al.*, 1998), and for investigating interactions between intonation and dialogue context with, for instance, context specific intonation models. For example, one can make different intonational predictions for a "no" answer to an unbiased information seeking *query y/n* and a "no" answer to a *check* question that expects a "yes". Kowtko(1996) finds different distributions for intonation patterns of acknowledgements in different sorts of games.

In considering what would constitute an improved dialogue act set, there are at least two directions one might take. One would be based on the functional role of dialogue acts in human conversation and computer dialogue systems. Act classifications would be judged on their psychological validity and/or explanatory power in dialogue analysis. The task would be to discover what formal properties of the acts, such as intonation or word order, could be exploited in the manner we have used here. Identification of the acts would also be an end in itself for dialogue systems, which might indeed be able to tolerate speech recognition errors to a certain extent as long as they understood what acts were being performed. The distinction between "yes", "yeah" and "yep" is not crucial to a system that has just asked a yes/no question.

Another direction would be to simply look for ways of classifying utterances that were useful for improving speech recognition results. For instance, one might iterate automatic training and recognition, perhaps in combination with an automatic clustering technique, to find a set of acts that gave optimal recognition results. There would be no guarantee that the resulting set would then be meaningful in dialogue terms, but if the goal is just improved speech recognition, that would not necessarily be a drawback.

References

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

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.

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 Deprivation and Drug Treatment. *In: Proc. 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.

Carletta, J., Dahlback, N., Reithinger, N., & Walker, A. 1997a. Standards for Dialogue Coding in Natural Language Processing. *In: Dagstuhl Seminar Report #167,Schloss Dagstuhl, D-66687 Wadern, Germany*.

Carletta, J., Isard, A., Isard, S., Kowtko, J., A. Newlands, A., Doherty-Sneddon, G., & Anderson, A. 1997b. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, **23**, 13–31.

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.

Eckert, W., Gallwitz, F., & Niemann, H. 1996. Combining stochastic and linguistic language models for recognition of spontaneous speech. *Pages 423–426 of: Proc. ICASSP '96*, vol. 1.

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

KOWTKO, J. C. 1996. *The Function of Intonation in Task Oriented Dialogue*. Ph.D. thesis, University of Edinburgh.

LADD, D. R. 1996. *Intonational Phonology*. Cambridge Studies in Linguistics. Cambridge University Press.

LEWIN, I., RUSSELL, M., CARTER, D., BROWNING, S., PONTING, K., & PULMAN, S. 1993. A speech-based route enquiry system built from general-purpose components. *Pages 2047–2050 of: EUROSPEECH 93*.

O'CONNOR, J. D., & ARNOLD, G. F. 1973. *Intonation of Colloquial English*. 2 edn. Longman.

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: ICSLP94*, vol. 1.

POWER, R. 1979. The organization of purposeful dialogues. *Linguistics*, **17**, 107–152.

RABINER, L. R., & 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: ICSLP96*.

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: EUROSPEECH 95*.

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.

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? *Submitted to Language and Speech (this issue)*.

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 ICSLP92*, vol. 2.

TAYLOR, P. A. 1995. The Rise/Fall/Connection Model of Intonation. *Speech Communication*, **15**, 169–186.

TAYLOR, P. A. 1998. Analysis and Synthesis of Intonation using the Tilt Model. *forthcoming*.

YOUNG, S., JANSEN, J., ODELL, J., OLLASON, D., & WOODLAND, P. 1996. *HTK manual*. Entropic.

Appendix - Computing the most likely move sequence

We show here the assumptions and approximations made in computing the most likely move sequence on the basis of the intonation model, the dialogue model and the speech recogniser. As mentioned in the body of the text, the computation is actually performed by Viterbi search. For the sake of simplicity, the role of the empirically determined weights is ignored here.

Notation

D	the dialogue
N_U	the number of utterances in D
C	cepstral observations for D
F	intonation observations, such as F_0
M	the sequence of move types for D
S	the sequence of speaker identites for D

Move Indentification

We want to find the most likely move type sequence $M^\star$, given speaker identities, cepstral vectors and intonation by solving:

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

Assuming that S, C and F are independent:

$$\begin{aligned} &= \underset{M}{\operatorname{argmax}} P(M)P(S|M)P(C|M)P(F|M) \\ &= \underset{M}{\operatorname{argmax}} P(S)P(M|S)P(C|M)P(F|M) \end{aligned}$$

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

$$= \underset{M}{\operatorname{argmax}} \quad \underbrace{P(M|S)}_{\substack{\text{dialogue}\\ \text{model}}} \quad \underbrace{P(C|M)}_{\substack{\text{speech}\\ \text{recogniser}}} \quad \underbrace{P(F|M)}_{\substack{\text{intonation}\\ \text{model}}} \qquad (6)$$

We assume that speaker identity has no effect on cepstral or intonational observations. This is clearly false, but we already make this assumption in using the same speech recogniser and intonation recogniser for both giver and follower. It should be clear from the discussions of the dialogue and intonation models that they compute the first and third terms of (6) respectively. We now show that the middle term of equation 6, $P(C|M)$, is in fact the contribution of the speech recogniser.

Letting W range over all possible word sequences,

$$\begin{aligned} P(C|M) &= \sum_W P(C|W)P(W|M) \\ &\approx \max_W P(C|W)P(W|M) \end{aligned} \tag{7}$$

where the replacement of summation by maximisation is a change from total likelihood to maximum likelihood. The value of W that maximises (7) is of course the sequence of words that will be the result of speech recognition.

Let

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

Now the two terms in equation 7 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 type specific language models.

Andreas Stolke (personal communication) suggests replacing the approximation in (7) by a sum over an N-best sentence list from the speech recogniser. This is obviously a closer approximation than made here but does require the recogniser to produce N-best lists, which can be time-consuming.