USING PROSODY TO ASSIST IN THE UNDERSTANDING OF SPOKEN ENGLISH

C. Rowles, X. Huang, M. de Beler [1]
J. Vonwiller, R. King, C. Matthiesson, P. Sefton and M. O'Donnell [2]

[1] Telecom Research Laboratories

[2] Sydney University

ABSTRACT - The use of prosody to assist in the understanding of spoken English by computers has recently started to attract some attention. While prosody has been studied for its potential in syntactic analysis and the understanding of dialogue structure, the practical use of prosody has been largely limited to improving the intelligibility of synthesised speech. In this paper we show how prosody can be used to improve syntactic segmentation and dis-ambiguation, assist in the understanding of dialogue structure and allow the proper management of turn-taking in dialogues.

INTRODUCTION

Although the role of prosody in spoken English has long been studied by linguists, it is only recently that the use of prosody has been seriously considered in the automatic understanding of spoken language by computers. In dealing with spoken English, we encounter several problems in recognition and understanding that are not apparent in written English. We do not, for example, have punctuation to segment utterances into phrases and sentences, and then assist in interpreting syntactic or semantic ambiguities. Prosody, however, can assist in solving these and other problems with spoken language.

In this paper, we present a model of prosody applicable to the understanding of spoken English, information-seeking, telephone dialogues. We have deliberately constrained our work to a simple domain to permit the implementation of a complete spoken dialogue understanding system. The model focusses primarily on pitch and pause information, but we show how this generalises to other features. This model is based on a systemic-functional model of dialogues that integrates phonology, grammar, semantics and context, and shows how prosody is used at each level of representation.

We then describe an experimental spoken language understanding system that takes its input as an unsegmented stream of words from a speech recogniser, segments it using prosodic and syntactic information and parses the stream into discourse and propositional moves. The parser uses a prosodic annotation of the input perfomed by a prosodic feature extractor to perform syntactic segmentation and disambiguation. The parsed moves are then analysed to determine speech function, dialogue role and propositional content, prior to querying a database. Prosody is used to assist the determination of speech function and extract the speaker's propositions from the unfolding dialogue. A dialogue manager controls the dialogue, using prosody to manage turn-taking and generating responses to the speaker based on the dialogue and information retrieved from the database.

Using examples (based upon recorded information-seeking dialogues between callers and an operator) of how the spoken language understanding system operates, we show that prosody can substantially improve the speed and accuracy of the parser in interpreting spoken phrases and sentences, and how prosody allows the tracking and understanding of dialogues that include sentence fragments and indirect or incomplete speaker requests.

A MODEL OF SPOKEN DIALOGUES

getting computers to understand and participate in spoken language dialogues over telephones, especially where utterances are likely to be terse, fragmented and ungrammatical, is a difficult endeavour. Added to this however, are the errors produced by attempting to recognise what words the speaker actually uttered. Given these difficulties, our approach has been to use as much linguistic knowledge as possible to improve understanding robustness. For this to be manageable, a model was required to integrate the phonological, grammatical, semantic and pragmatic levels of spoken dialogues. We chose a systemic-functional model of information-seeking dialogues developed at Sydney University (Eggins et al, 1991). While this has the disadvantage that it is complex, it has the advantage of providing the level of linguistic integration we sought, and the complexity is manageable in our relatively simple domain.

The model represents dialogues between two agents (the caller is one, the system is the other) as a set of communicative states and communicative actions (usually conversational moves, but may also be non-linguistic) that move the dialogue from one state to another. From any given state there are many possible actions depending on the communicative context of that state, and each action has its own activation conditions and its own effects, which are new dialogue states. The model is dynamic, and can thus handle the logogenetic unfolding of a dialogue in time (O'Donnell 1990).

Each state in a caller-system dialogue has a generic domain context (i.e. what the call is about) and an interactional context (where are we in the exchange), and each action has an interactional context and a speech function (what an agent is trying to achieve with an individual conversational move). Moves are realised within lexico-grammatical and phonological bounds. These contexts are reflected in the hierarchical linguistic strata of the systemic-functional model. The model integrates these contexts and links them to specific moves via taxanomic inheritance networks. There are networks for Generic Context, Exchange Context and Speech Function. Each network provides a taxonomic classification of moves within that context. Figure 1 shows a fragment of the Speech Function network. Network branches provide alternative (illustrated) or combined (not illustrated) choices. In the Generic Context Network, combined choices provide, in effect, sub-goals that must be completed to finish an information seeking dialogue. The Exchange Context Network provides a classification for turn-taking, initiation and completion within exchange moves, while the Speech Function Network classifies moves according to their role within an exchange.

During a dialogue each move from the caller is analysed and classified according to each taxonomic network. At each branching point in a network (called a system), attributes of the current move and the current dialogue state are compared to matching conditions for the next lower classifications (features) to select the most appropriate to describe that move. The Speech Function Network uses move attributes such as mood, prosody and type (e.g. declarative) to classify the speech function of the move within the dialogue and determine the caller's proposition.

Once the speech function of a move is known, the Generic Context Network uses the propositional content of the move (i.e. query specification) to classify the move according to which sub-goals have

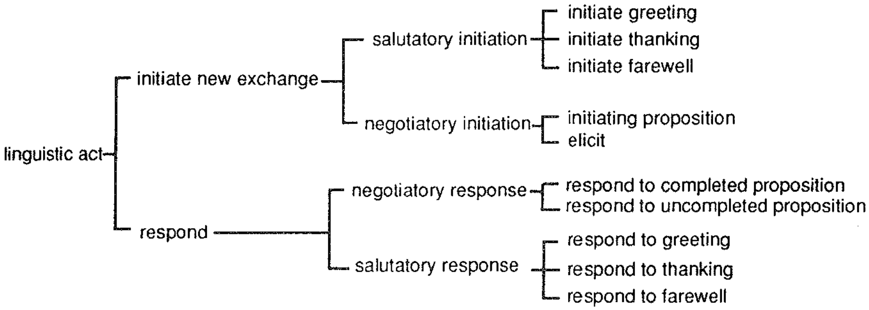

Figure 1. Fragment of Speech Function Taxanomic Network

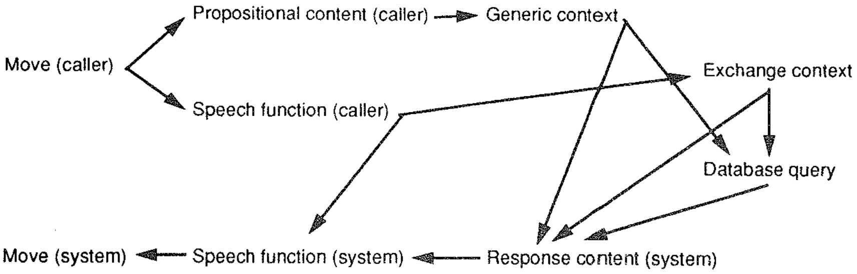

Figure 2. Information flow during analysis and generation.

been completed and what information is still required. The Exchange Context Network then uses the speech function and tone contour to update the current state of the dialogue, such as who initiated the current exchange and who is expected to continue the dialogue. If the caller's query is complete, and the caller does not wish to change it, a database may be searched for the required information. If not complete, the Exchange and Generic Contexts indicate how the system should respond, such as asking for more information or waiting for the caller to complete the query. If the system should respond, the Exchange and Generic Contexts are used to determine the type of response required and the Speech Function Network is used to determine the attributes (i.e. mood, tone contour, etc) of the system move to be generated. The tone contour representation is basically that of (Halliday 1985).

The model can thus, be used to analyse a caller's moves as well as generate appropriate responses. The sequence of actions is shown in Figure 2.

We have implemented this model in a system for understanding spoken English. At present, the model addresses the analysis and maintenance of spoken, information-seeking, dialogues. Thus, while the model uses lexical and prosodic information in its analysis, these are generated by a prosodic feature extractor and a parser that translate the moves' surface form into a logical representation.

USING PROSODY DURING PARSING

The parser takes its input as a recognised stream of words, annotated with prosodic features. A parser pre-processor takes the word string together with pitch markers and pauses, annotating the word string with pitch markers (low marked as "~", medium "-" and high "^") and pauses (short "*" and long "**"). Pitch levels are relative to the pitch range. Short pauses are less than 250ms. The pre-processor uses the pitch and pause markers to segment the word string into intonationally-consistent groups, such as tone groups (boundaries marked as "<" and ">") and moves (//). It also locates fixed expressions so that during the parsing nondeterminism can be reduced and uses tone group information to help resolve possible fixed expression ambiguity ("Mary helped John to look after his kids" vs "I'll look after you do").

The parser is written in the Definite Clause Grammar formalism and enhanced by semantic routines. It is capable of resolving most of the ambiguous structures it encounters in parsing written English sentences, such as coordinate conjunctions, pre-positional attachments, and lexical ambiguity.

Moves input to the parser are unlikely to be well-formed sentences. The parser first assumes that the input move is lexically correct and tries to obtain a parse for it, employing syntactic and semantic relaxation techniques for handling structurally ill-formed sentences. If no acceptable analysis is produced, the parser asks the speech recognizer to provide the next alternative word string.

The output of the parser is a parse tree that contains syntactic, semantic and prosodic features. Most ambiguity is removed in the parse tree, though some is left for later resolution, such as definite and anaphoric references, whose resolution normally requires inter-move inferences. The parser also detects cue words in the input using prosody.

During parsing prosodic information is used to help disambiguate certain structures which cannot be disambiguated syntactically/semantically, or whose processing demands extra efforts without prosody. The prosodic markers are used as additional pre-conditions for grammatical rules, discriminating between possible grammatical constructions via consistent intonational structures. Currently the following types of ambiguity are handled with prosodic aids: homographs, fixed expressions, prepositional phrase attachment, and coordinate constructions.

In order to process fixed expressions, in the system's fixed expression lexicon, we have entries such as "fix_e([gave, up], gave_up)". The pre-processor then contains a rule to the following effect, which conjoins two (or more) words into one fixed expression only when there is no pause following the first word:

```
match_fix_e([FirstW, SecondW|RestW], [FixedE|MoreW]):-
    no_pause_in_between(FirstW, SecondW),
   fix_e([FirstW, SecondW], FixedE),
   Match_fix_e(RestW, MoreW).
```

This rule produces the following segmentations:

(1) <~He ~gave> *<^up to ^two hundred dollars> *<~to the ^charity>**//

(2) <~He ^gave ^up> *<^two hundred dollars> *<-for damage compensation>**//.

In (1), *gave* and *up* to are treated as belonging to two separate tone groups, whereas in (2) *gave up* is marked as one tone group. The pre-processor checking its fixed expression dictionary will therefore convert *up to* in (1) to up_to, and *gave up* in (2) to gave_up.
Similarly we use pauses to help resolve pp attachment:

(3) <I would like> < information on her arrival> [no pause present between "information" and "on hter arrival" - "on her arrival" attached to "information"]

(4) <I would like> <information> ** <on her arrival> [pause present after "information" - "on her arrival" attached to "like"]

Pitch information is also incorporated into the grammar. The following is a simplified version of the VP grammar to illustrate the usage of both pitch and pause information:

```
/* Verb phrase rule 1.*/
Vp --> V_intr.

/* Verb phrase rule 2. Some semantic checking is carried out after a transitive verb and a noun
phrase is found.*/
Vp --> V_tr, Np, {match(V_tr,Np)}.
/* If a verb is found which might be used as intransitive, check if there is a pause following it.*/
V_intr --> [Verb], {is_intransitive(Verb)}, Pause.
/* Otherwise see if the verb can be used as transitive.*/
V_tr --> [Verb], {is_transitive(Verb)}.

/* Noun phrase rule.
"Mods" can be a string of adjectives or nouns: major (races), feature (races), etc.*/
Np --> Det, Mods,HeadNoun.
/* Head noun is preferred to be low-pitched.*/
```

```
HeadNoun --> [Noun], {low_pitched(Noun)}.
/* This succeeds if a pause is detected. */
Pause --> [pause].
```

Compared with parse times for moves without prosody, the prosodically annotated, ambiguous moves are parsed significantly more quickly with these prosody rules, while unambiguous moves suffer only slight increases in parse times due to extra rule execution (Rowles & Huang 1992). In some cases, moves were only parsed correctly with prosody. In both cases, prosody is shown to be an important factor. This is true whether the input is a sentence or sentence fragment, as tone contours can indicate difference in single words intended as clarification questions or confirmations.

DIALOGUE UNDERSTANDING & MANAGEMENT

The dialogue analyzer accepts the parsed moves as input and tries to determine speech functions, dialogue roles and propositional content. Prosody is also used in the process to assist the determination of speech function and extract the speaker's propositions from the unfolding dialogue.

The Dialogue Analyser implements the model described earlier. Speech function is determined from the Speech Function Network, which uses the move's tone contour as one of the classification attributes. At the dialogue level, long pauses are used to separate moves and, via the Exchange Context Network, determine whose turn it is next. Tone contours are used in dialogue analysis, rather than the parser'sprosodic features, as they more directly relate to the caller's intentions (Halliday 1985).

An interesting aspect of the systemic model is that way in which sentence fragments are analysed. A partially recognised or terse utterance such as a location name can still be correctly understood in terms of its dialogue context, as the tone contour can allow a final choice between the small number of speech functions or exchange contexts possible at any point in a dialogue. Thus, for example, a caller may finish a query, with a weak termination tone contour, wait, and then utter a single word, such as a location. When it sees a weak termination, the dialogue analyser will form the logical database query, but then wait to see if the caller will say anything else. A subsequent declarative, such as a location name, will then be recognised as a continuation of the caller's turn with a change of a query parameter.

The Dialogue Manager controls the dialogue, calling the Dialogue Analyser whenever a move is received from the parser. The Dialogue Manager uses pauses and the tone contour towards the end of a move, to help manage turn-taking and generating responses to the speaker based on the dialogue and information retrieved from the database. The Dialogue Manager also implements some minimal strategies for dialogue repair and recovery, although more work is required.

The Dialogue Manager receives moves from the post-parser, with attributes instantiated as much as possible. Analysis proceeds via the sequence shown in Figure 2, considering the new move in terms of its dialogue and task contexts. The linguistic content (mood, tone-contour and speech-function) of the current move is matched against the possible branches in the Speech Function network constrained by context of the previous move, to obtain the most likely speech function classification. We then apply a set of rules which give new exchange context features based on a list of speech function attributes. For example if the prosody of the caller's move indicates a "weak-termination" attribute, the corresponding exchange context attribute will be "speaker-may-continue".

Throughout the dialogue, exchanges are initiated, suspended, returned to and abandoned. We thus need to keep some kind of exchange history: each exchange context is stored in a stack, the current one being at the top, and any exchange can be accessed by the system at any time. The exchange context attributes tell us whose turn it is next, who initiated the current exchange, what type of exchange it is (negotiatory, greeting, thanking, etc.). Symmetrically, a list of possible speech function attributes for the next move can be created: it contains all speech function attributes which have one of the new exchange context attributes as an activation condition. We now update the generic context attributes, which provides information about the overall agenda for the dialogue, i.e. what the caller and system must achieve. The Generic Context Network is different from the other two in the sense that the attributes are incrementally gathered throughout the conversation, not a new set for each new move. The interface between the exchange and generic contexts is performed through a set of generic tasks (e.g. greeting, location specification, request another service, etc.). Tasks still to be done are indicated by the Generic Context Network, while the exchange context determines when a task may be done. The contextual effects of the current task give us the new generic context attributes. The tasks with true activation conditions are now members of the new system's or caller's do-able tasks list.

By classifying each speaker's utterance in terms of the system networks, we have the tools to perform the other functions of the Dialogue Manager. Because of the symmetry of the systemic grammars, the system's move generation and speaker's move analysis can be dealt with in a similar fashion. The turn management function uses the current exchange context attributes to decide if turns should be swapped or not. If the caller is not expected to continue, the turn is given to the system. The order of operations necessary to generate a system move is the opposite one as for analysing a caller's move: the generic task has to be set first. The previous generic context attributes gave us a list of machine do-able tasks (e.g. greet, ask for location specification, start database search, etc.). Each task has a priority and activation conditions: they tell us for example which combination of query parameters are sufficient to commence a database search. Once we know the generic task, the generic context and exchange context attributes can be updated (like for a caller's move). A system move is also created: it has the same structure as the moves generated by the post-parser. The prosody of the move is used by the speech generator to decide if the content should be said as an affirmation or a question, and what is the preferred intonation. This area has yet to be implemented.

CONCLUSIONS

We have presented a systemic-functional model for the understanding and generation of conversational moves in an information-seeking, spoken dialogue setting, and have shown how the use of prosody aids understanding. The use of hierarchical taxonomic networks to represent various linguistic strata allows the integration of a wide range of linguistic resources and these resources to be used at various stages of analysis and generation. Indeed, their symmetry allows us to navigate in both directions: speech function to exchange context to generic context for the analysis of a caller's move, and generic context to exchange context to speech function for the generation of a system's move. We have shown how prosody is represented in this model and how it assists in syntactic, semantic and dialogue analysis, and in dialogue management. The system has been implemented and demonstrates the value of a systemic approach and the use of prosody.

ACKNOWLEDGEMENTS

The permission of the Director of Research, Telecom Australia to present this paper is gratefully acknowledged.

REFERENCES

Eggins, S., Vonwiller, J., Matthiesson, C., and Sefton, P., (1991), *Analysis and Models of Information Seeking Dialogues*, 18th International Systemic Congress, Tokyo, 1991.

O'Donnell, M. (1990), *A Dynamic Model of Exchange*, Word, Vol. 41, 3, December, 1990.

Halliday, M.A.K., (1985), *Introduction to Functional Grammar*, Edward Arnold, London.

Rowles, C.D. & Huang, X. (1992) *Prosodic Aids to Syntactic and Semantic Analysis of Spoken English*, 30th Annual Meeting of the Association for Computational Linguistics, Newark, Delaware, June-July.