GENERATING PITCH ACCENTS IN A CONCEPT-TO-SPEECH SYSTEM USING A KNOWLEDGE BASE

Sandra Williams

Microsoft Research Institute,
School of MPCE, Macquarie University, NSW 2109, Australia.

ABSTRACT

This paper describes a concept-to-speech system for generating spoken descriptions of routes between places within Macquarie University Computing Department. The Natural Language Generation (NLG) component of the system generates a textual route description marked with intonational information. The discourse structure of the route description is related closely to the knowledge representation of the route. The NLG component includes a pitch accenting algorithm which places appropriate pitch accents on elements of the utterance requiring particular emphasis or stress.

Our pitch accenting algorithm uses a domain knowledge base and a discourse history. From these it determines whether information selected to form the content of the utterance is shared mutual domain knowledge, given information, or new information. It can then assign an appropriate pitch accent to one word in each prosodic phrase. The text-to-speech component then determines the appropriate syllable to be accented in the word.

1. INTRODUCTION

Concept-to-speech is a term used to describe systems that integrate the technologies of Natural Language Generation (NLG) and Text-To-Speech (TTS) with the aim of improving the intelligibility of generated speech. Contemporary TTS systems tend to read text in a way that sounds unnatural. This is due partly to deficiencies in syntactic analysis of raw input text, but also to lack of semantic information and world knowledge. Some TTS systems accept input text pre-marked with intonational information. Such inputs can make a great deal of text pre-processing unnecessary. In concept-to-speech systems, intonationally marked textual input is automatically produced by NLG. Recently concept-to-speech has begun to emerge as a new technology and number of research systems now exist.

Marking the placement of pitch accents on syllables requiring particular emphasis or stress in an utterance, the subject of this paper, is just one issue in generating intonationally marked input for TTS system. Previous work on the placement of pitch accents includes Davis and Hirschberg [1], Theune [2], Prevost [3], and Hiyakumoto et al [4]

Assignment of pitch accents requires not only syntactic structure, and information structure, but also world knowledge. The use of world knowledge to assist with placement of pitch accents was suggested by Theune, who rejected it in favour of a data structure approach. Hiyakumoto et al. used a large-scale semantic lexicon as a source of world knowledge, but found that this could not cope with all relationships necessary for pitch assignment. A common problem cited with knowledge bases is that building them is too labour intensive. However, we have demonstrated that a large knowledge base can be built *automatically* from an existing database [5] thus cutting down considerably on the amount of work involved. The concept-to-speech system described in this paper uses a small hand-constructed knowledge base for experimenting with the use of a knowledge base for pitch accent assignment. However the same techniques could be used with a much larger automatically generated knowledge base.

This paper describes a preliminary implementation of our concept-to-speech system. The NLG part of the system assigns pitch accents to words in utterances that it identifies as requiring particular stress or emphasis. The correct syllable to be accented within a word is identified by the TTS system. Like Hiyakumoto et al, we assign two kinds of pitch accents: H*, and L+H*. These accents are part of the ToBI intonational marking scheme [6]. We assign L+H* to utterance elements relating to previous parts of the discourse, including given information and contrastive elements. We assign H* to new information.

The ideas behind our implementation were motivated by data from a pilot corpus collection. We recorded speech data from a number of people describing routes between different locations in the Computing Department. This pilot corpus is small, but it contains examples of the kinds of utterances we attempt to generate, obtained from a number of different speakers.

Here is a sample of output from our system:

```
Walk down the corridor. On the wall straight
                L+H*
ahead of you, you will see the MRI noticeboard.
  H*                              H*
When you reach it, turn left.
               L+H*         H*
```

In the first sentence 'the corridor' is mentioned for the first time, but it is not treated as new information. Where this phrase occurs in our pilot corpus, 'the corridor' was not spoken with stong accenting. We hypothesize that in the domain of route descriptions within a building, features that are part of the building are treated as common knowledge. The use of the definite determiner 'the' also suggests shared mutual knowledge, rather than introduction of previously unknown information. Similarly with 'the wall'. Our system 'knows' that corridors are parts of buildings, and walls are parts of corridors through its knowledge base object hierarchy. 'Corridor' is assigned the 'given' L+H* accent, but 'wall' is deaccented in favour of 'ahead' (see Section 4 for an explanation). Directions, such as 'straight ahead' and 'left' are generally treated as new information, but

may be contrastive. Our system only assigns accents to noun phrases at present. In the final sentence of our example, assigning the L+H* accent to the verb 'reach', rather than the noun phrase 'it' might have given more natural sounding output.

2. SYSTEM OVERVIEW

Our current demonstration system generates spoken language descriptions of routes between places in the Department of Computing. The system can be used in a scenario where a stranger who is visiting the department needs to find his/her way to a particular person's office, or to a place such as the seminar room. The system could be located in a fixed information point at the entrance to the department, or in a hand-held device. It is accessed via a Web browser that displays a floor plan of the Department of Computing shown in Figure 1. Two pull-down menus allow the user to select a start point and end point of the proposed route.

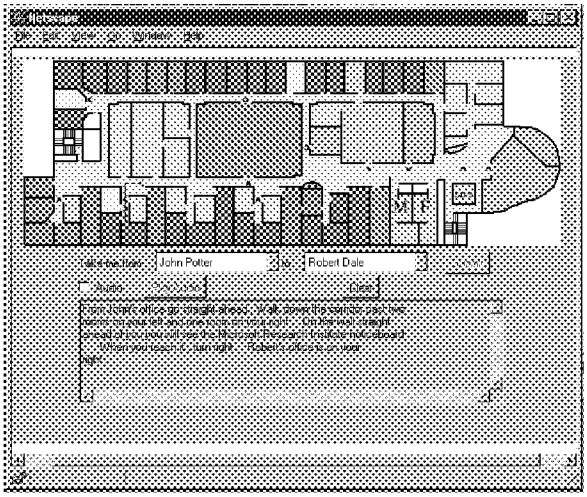

Figure 1: Screen shot of the multimodal user interface.

A check box to request audio file output and a 'Play Audio' button enable the spoken description of the route to be played out. The route is displayed as a line on the floor plan. The text of the spoken route is displayed in the text field.

A system such as this requires a number of components. The user interacts with the system via the Web interface where the graphical, textual and audio versions of the route can be accessed. The user's input is converted into a route by a route planner. The route is rendered into text and speech by the concept-to-speech system consisting of an NLG module and a TTS module and their associated Knowledge Bases and Discourse History. Interprocess communication is through a Dialogue Manager. These major system components are shown in Figure 2. A very brief description of each module is given below:

Multimodal Interface. The system is accessed via a Web browser, which starts up a Java applet[1]. It controls the display of the map, text field, pull-down menus, buttons, check box and audio file access. It sends the user input to the route planner and the route planner output to the Dialogue Manager.

Dialogue Manager. The Dialogue Manager's function is to facilitate inter-process communication. It consists of a Prolog server that communicates with the Java applet and with the Festival server via sockets and with the Accenting Natural Language Generator directly in Prolog.

Route Planner[1]. The input to the Route Planner is the start point and end point of the route selected by the user. The planner represents the map internally as a graph of nodes (or symbolic points) and edges with distances. It plans the shortest path between the start and end nodes using a standard branch-and-bound search algorithm. The result is as a list of nodes.

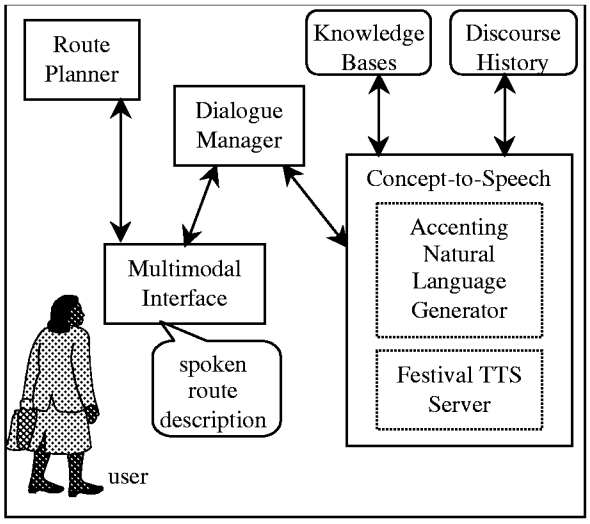

Figure 2: System Architecture.

Knowledge Bases. The object knowledge base models the domain of building interiors. It contains an object hierarchy linked by 'is a' and 'part of' relationships. These are the fundamental 'concepts' of the concept-to-speech system. This hierarchy is essential for both the text generation and pitch accent processing. The hierarchy is implemented as a set of Prolog facts, e.g. 'c4 is a corridor', 'w4E is a wall', 'w4E is part of c4': `isa(r332, room). isa (c4, corridor). partof(w4E, c4).` Objects have properties associated with them, e.g. directions and connections to other objects. The object hierarchy and object properties are used to work out relationships between objects in the domain. So we can work out what something is, what its properties are, what it is part of, and what is part of it.

Another knowledge base contains general spacial knowledge, such as 'if you are heading North and you want to go East, then

[1] This was developed by Derek Santibanez, an Honours student at Macquarie University.

you must turn right'. Additional knowledge bases contain the lexicon and grammar for NLG.

Discourse History. This contains items mentioned in the previous discourse. At present it contains only noun phrase items. The Discourse History is used in the generation of referring expressions and in the generation of pitch accents.

Accenting Natural Language Generator (ANLG). This component is one half of the concept-to-speech module. It generates a natural language description of the route annotated with pitch accents for the Festival text-to-speech system.

This module consists of the components summarised below:

- Content Selector. This transforms a simple list of points into a series of route sections (corridor sections) and turns. For each corridor section, salient landmarks and features are selected. On selection, these items may be marked for pitch accenting (see Section 4).
- Sentence Planner. Currently the text to be generated consists of a series of templates. Templates are assigned according to the structure of the route representation. The slot fillers for these templates consist of syntactic categories and associated data from which phrases can be generated by the surface realiser.
- Surface Realiser. This generates the phrasal slot fillers for the sentence templates using the lexicon and grammar. Where appropriate, ToBI annotations for pitch accents are generated.

Festival Text-To-Speech Server. Festival was developed at Edinburgh University [7]. This component is the other half of the concept-to-speech module. Festival is used in server mode and ToBI input mode. Festival generates an audio file which can be accessed by the Web browser and played out on the user's machine.

3. ROUTE REPRESENTATION AND CONTENT SELECTION

A crucial aspect of our Language Generation process is that it relates discourse structure to the structure of the route representation. In this section we will explain how the route representation is built, how it is augmented with additional information, and how items to be mentioned in the route description discourse are selected.

3.1. Route Representation

Our route representation is similar to representations used by [1]. It consists of a series of corridor sections and turns:

```
routeRep = section, (turn, section)*
section = start,via,end
start = pStart
via = [p1,p2,p3,...,pn]
end = pEnd
turn = lhs | rhs | back
```

The '*' and parentheses represent optional elements occurring zero or more times. p1, p2, pStart, pEnd, etc. are point symbols. Corridor sections are represented by start and end points and a list of intermediate points. Points can be either junctions between corridors (e.g. T-junctions, crossroads and corners) or 'orientation points' e.g. outside rooms, where one would have to change direction to reach the room, or look in that direction to see it. These orientation points have the property that they connect to something in a particular direction. Some orientation points connect to landmarks such as signs hanging from the ceiling. Some points are both junctions and orientation points (e.g. where there is a landmark at a junction).

To build the route representation, the Content Selector uses the knowledge base to work out where turns should be inserted. It transforms the list of points received from Route Planner into a route representation. For example the input list `[p331,p333,p332,p330,p2,p328]` becomes (part deleted due to lack of space):

```
section(start(p331),via([p333,p332,p330]),
end(p2)), turn(rhs), section(start(p2), via([]),
end(p328))
```

Note that the first and last sections connect orientation points to rooms which are 'actual places' where the route starts and ends, in addition to the orientation points in the input.

'Via' lists are further processed to replace orientation points with places they are connected to and add directional information. Orientation points connected to rooms, etc, are annotated with either 'lhs' (left hand side) or 'rhs' (right hand side). The symbolic name for the corridor is also inserted. For example, the first 'via' list above becomes:

```
via([c4,r333:lhs,r332:lhs,r330:rhs])
```

where c4 is the corridor symbol, r333, r332 and r330 are room symbols.

3.2. Content Selection

Up to this point, the Content Selector has expanded the representation with additional information and substituted rooms for points. After this, it selects information to be included in the discourse. The issue of what to include and what to exclude in an optimum route description is problematic since the notion of what an optimum route description actually is, is undefined. The present Content Selector picks out information indicated by our pilot corpus collection and from our own intuitions about what should be mentioned. The 'via' list above becomes:

```
via([c4,pass(lhs,2,room),pass(rhs,1,room),final(
n101,w1N,ahead)])
```

The following data has been selected:

Content of corridor sections. This is reduced to a count of rooms passed on both sides, and a salient landmark, if one exists, or if not, a final room to comment on at the end of the section. In the example above, 2 rooms have been found on the LHS and 1 on the RHS. These are denoted by pass(lhs,2,room) and pass(rhs,2,room). final(n101,w1N,ahead) means that landmark n101 (the MRI noticeboard) has been found on wall w1N in the direction of travel.

Landmark selection. Landmarks are selected by a set of common-sense rules derived from a preliminary analysis of the way landmarks are utilised in our pilot corpus:

- Choose landmarks located at the end of corridor sections.
- Salience is determined by position and direction of travel (e.g. 'ahead', located on the wall straight ahead in the direction of travel, takes precedence over 'down' on the floor, followed by 'up' on the ceiling, then on the left/right walls of corridor).

Many other factors could potentially be taken into account e.g. length of the corridor and optimum number of items to mention.

4. PITCH ACCENT PLACEMENT

Our assignment and placement of pitch accents is motivated by examples from our pilot corpus. The core of our pitch accent assignment algorithm is a set of decisions on whether entities (nouns phrases only at present) selected by the Content Selector contain given or new information, or whether they contain shared mutual knowledge.

All items relating to noun phrases previously mentioned in the discourse are stored in a discourse history list. Suppose the items c4, n101, and w1N from the 'via' list example in the previous section are to be evaluated for accentuation. First c4 , the corridor, is checked. The history list does not contain c4 yet, because this is the first time it has been mentioned. There is the possibility that it could be assigned an accent equivalent to that of a new item (H*). The object hierarchy is consulted, and we find that c4 is a corridor which is a building part. It is assumed that corridors are common mutual knowledge and c4 is thus assigned an accent equivalent to that of a given item (L+H*). The wall, w1N is processed in exactly the same way. n101 on the other hand, retains H* because it is a landmark, not a building part.

Items are grouped into prosodic phrases by the Sentence Planner (at present these phrases are fixed in our templates). Since we only want to select a single pitch accent per prosodic phrase, if there is more than one accented item, those with the H* accent are preferred over those with L+H*. If there is more than one item with H*, our present solution is to select the final one occurring in the prosodic phrase.

In the above example, c4 is assigned to the template to generate the sentence: 'Walk down the corridor'. It retains its 'given' accent of L+H* since it is the only accented item in that phrase. In the case of the wall, w1N, it is assigned to a phrase, 'On the wall straight ahead of you '. This contains a direction which is automatically given an H* accent equivalent to a new item. In this phrase, 'the wall' is deaccented. In our system, direction properties (realised as 'left', 'right' etc.) are always assigned accents in preference to other items. They are either treated as new information (H* accent), or contrastive information (L+H* accent). Contrast is derived from similar data structures, following Theune [2]. The '2' and the direction 'left' are both contrasted in the following phrase:

```
3 rooms on your right and 2 rooms on your left.
H*              H*      L+H*            L+H*
```

After the Surface Realiser has generated the output text marked according to the ToBI intonation scheme, it is input to Festival. Festival is accessed in a mode with default intonation turned off and ToBI enabled. In this mode unmarked text would be spoken in a monotone (although word-level stressed and unstressed syllables are still produced correctly).

5. CONCLUSIONS

The pitch accenting algorithm works well in the domain of the application. This work shows that information about mutually shared knowledge derived from a knowledge base, when used in addition to given/new information derived from a discourse history can be used to inform decisions on pitch accent placement.

6. ACKNOWLEDGMENTS

We would like to thank: Catherine Watson for advice on speech issues; Derek Santibanez for the Java Multimodal Interface applet and the Route Planner; Robert Dale for advice on Natural Language Generation and route representation with Kyle Dean, Maurice Pagnucco and Norman Foo; Alan Black for help with the Festival server; Stephen Green for Festival, Sicstus Prolog and Java installation; and members of the MRI Language Technology Group for their advice and encouragement.

7. REFERENCES

1. James Davis and Julia Hirschberg [1988] Assigning Intonational Features in Synthesized Spoken Directions. ACL'88.
2. Mariët Theune [1997] Generation of soccer reports: some results. ACL/EACL, Madrid, Spain.
3. Scott Prevost [1995] A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation. PhD Thesis. University of Pennsylvania.
4. Laurie Hiyakumoto, Scott Prevost and Justine Cassell [1997] Semantic and Discourse Information for Text-to-Speech Intonation. ACL/EACL Workshop on Concept-to-Speech Generation Systems.
5. Robert Dale, Stephen Green, Maria Milosavljevic, Cécile Paris, Karin Verspoor and Sandra Williams [1998] The Realities of Generating Natural Language from Databases. 11th Australian Joint Conference on Artificial Intelligence, 12-17 July, Brisbane, Australia.
6. K. Silverman, M. Beckman, J. Petrelli, M. Ostendorf, C. Wrightman, P. Price, J. Pierrehumbert and J. Hirschberg [1992] ToBI: a standard for labelling English prosody. ICSLP92, vol. 2, pp. 867-870.
7. Alan Black, Paul Taylor and Richard Caley [1997] The Festival Speech Synthesis System. Edition 1.2, for Festival Version 1.2.0. Edinburgh University, UK.