AUTOMATIC RECOGNITION OF INTONATION FROM F_0 CONTOURS USING THE RISE/FALL/CONNECTION MODEL

Paul Taylor
(email: paul@itl.atr.co.jp)

ATR Interpreting Telecommunication Laboratories, 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, JAPAN

ABSTRACT

This paper describes an automatic system for labelling intonational tune information based on the Rise/Fall/Connection model of intonation. The system is powerful in that it presupposes no prosodic knowledge of the utterance it is recognizing, and is capable of labelling all the intonational tune effects of English.

1. INTRODUCTION

This paper describes a system which can automatically extract a phonological description of an utterance's intonational tune from its F_0 contour. The phonological description of an utterance's intonation can be thought of comprising if four effects: *phrasing*, *pitch range*, *tune association* and *tune type* [5]. This paper is mainly concerned with describing where the pitch accents in a phrase occur (tune association), and with describing what form these pitch accents take (tune type). Pitch range is examined indirectly, but this system makes no attempt at labelling prosodic phrase boundaries. (A number of recent papers provide promising results to the solution of that problem e.g. [1], [8], [7].) The main application for the system described here is for high level linguistic processing in a machine interpretation system, but the system could also serve in a dialogue system or as a system for automatically labelling intonation for data analysis purposes.

1.1. Why Tune Knowledge is Important

Most speech recognition systems make little use of prosody, preferring instead to try and extract the text of an utterance from an analysis of that utterance's segmental content. Recently, some systems have started to use prosodic phrasing information to aid in parsing [7], while other have used stress and pitch accent information to help in the lexical lookup process [2]. Although the system presented here could be of use in these types of systems, the main reason for using intonational tune in speech recognition is because it can signal speech acts and intentions in the speaker which are not manifested in any other way in the utterance.

At ATR, we are concerned with automatic interpretation of English and Japanese. For correct interpretation it is not only necessary to discover the words that have been spoken, but also the underlying intention of those words, and much of this speech act information can be derived from intonational and specifically tune information.

For example, in a typical hotel booking dialogue, the receptionist may say: "Your room will cost 20,000 yen" and the guest may say "yes" in reply. Depending on the tune type of the "yes", a variety of different meanings can be imparted. A simple "yes" spoken with low falling intonation and not much strength could mean "I understand, continue"; a "yes" with high intonation and a rising tune would indicate surprise, maybe as the guest considers the hotel very expensive; and a "yes" spoken low in the pitch range with a rising intonation may indicate uncertainty, as if there is some undesirable condition associated with the room's price. These differences in speech act are primarily indicated by the utterances tune.

The study of how different tunes signal different intentions is not the subject of this paper; rather we are interested in being able to automatically derive the tune description of an utterance from its acoustic form, so that that the higher level linguistic aspects of the interpretation system have data to work with.

2. THEORY OF THE RFC MODEL

The *Rise/Fall/Connection* ("RFC") model of intonation was designed so as to provide a system of linking the F_0 and phonological descriptions of an utterance. The phonological tune description system (the "HLCB" system) and the method of labelling F_0 contours with that system are described in Taylor [5]. The HLCB tune description system is only one of many that exist for English (e.g. [3], [4]). While it is argued that the HLCB system has some purely linguistic advantages over these other systems [5], the main reason for its use here is that the HLCB system was specifically designed to be "formal" and therefore easily computable, whereas most other phonologies were not designed with this aim in mind.

The model makes use of an intermediate level (termed the "RFC" level, from which the name of the overall system is derived) which lies between the F_0 and phonological levels of description.

2.1. The Intermediate (RFC) Level and its Relation to F_0 Contours

The theory of the intermediate level states that any F_0 contour can be described by using a linear sequence of non-overlapping *elements*. There are three elements: *rise*, *fall* and *connection*. The equation (termed the *monomial function*) for the rise element is the same as the fall element equation relected in the y-axis. The connection ele-

f	(starts late in syllable)	$\mathbf{H}_d$
f	(starts early in syllable)	$\mathbf{L}_a$
r f	(late peak)	$\mathbf{H}_l$
r f	(high peak)	$\mathbf{H}_e$
r	(on accented syllable)	$\mathbf{L}$
r	(not on accented syllable)	$\mathbf{B}$
r	(non-phrase final)	$\mathbf{B}_i$
c		$\mathbf{C}$
c	(rising)	$\mathbf{C}_r$

Table 1. *A grammar to link RFC and HLCB descriptions*

ment is a straight line. In addition to be marked "rise", "fall" or "connection", each element has two scaling factors. The first number represents the duration of the element in seconds and the second the amplitude of the element in Hertz. The form of the equation using the scaling factors is given in equation 1, where f_0 represents F_0, t represents time, D is the duration of the element and A is the amplitude. γ is used to control the curvature of the rise and fall elements, and in principle this is variable. However, in practice a constant value of 2.0 was found to be adequate in all cases.

$$\begin{array}{ll} f_0 = A - AC.(t/D)^{\gamma} & 0 < t < D/2 \\ f_0 = A.C.(1-t/D)^{\gamma} & D/2 < t < D \end{array} \quad where \;\; C = 2^{\gamma - 1} \tag{1}$$

The RFC description is a complete comprehensive description of the F_0 contour and from this description contours which closely resemble the original can be resynthesized [6]. Thus all the intonational information of an utterance can be represented in an RFC description; so pitch range information could be extracted in addition to tune information.

2.2. Phonological Level and its Relation to the RFC Level

The phonological system has four axes of description (phrasing, pitch range, tune association, and tune type) of which we are only concerned with tune association and tune type here.

The domain of the syllable is used to describe association as no more than one accent can occur on a syllable: we simply describe association by saying that a tune element is associated with a particular syllable.

Four phonological elements are used to describe tune: **H, L, C** and **B**. Pitch accents are described as being of type **H** (high) or **L** (low), **C** is used to describe phonologically significant connection elements, and **B** is used to describe the rise elements that may occur at phrase boundaries. **H** elements are used to describe pitch accents which are manifested as peaks in the F_0 contour. Within this class the features "late", "downstep" and "elevated" are used to subclassify accents. **L** elements are use to describe pitch accents which are manifested as valleys in the F_0 contour, and a single feature "antecedent" is used to subclassify accents in this group.

The grammar described in table 1 is used to produce a HLCB description from an RFC description.

3. THE AUTOMATIC ANALYSIS SYSTEM

The automatic analysis system has three modules: the F_0 processor, which does some preliminary smoothing on F_0 contours, the RFC labeller which takes these contours and produces an RFC description, and the HLCB labeller, which produces a HLCB description from RFC information.

3.1. F_0 Processor

It is common in intonational analysis to use voiced speech that does not contain obstruent segments as F_0 contours from such utterances are smooth and continuous. In these utterances the F_0 patterns of the intonational tune are easier to detect and describe using visual analysis than contours extracted from an unrestricted segmental environment. The RFC labeller (see below) also works best on this type of data.

In practice, we want to be able to analyse any F_0 contour, not just those from ideal segmental environments. The F_0 processor module was built to normalise for the unvoiced regions and obstruent effects in F_0 contours, and produce contours which are more like the smooth continuous ones.

The F_0 processor takes "raw" F_0 contours specified in 5ms frames and performs 15 point median smoothing, which largely reduces the influence caused from obstruents. This technique is not entirely successful as the resultant contours are still somewhat affected by the presence of obstruents. Heavier smoothing could remove these effects completely, but if the smoothing is too heavy, the intonational content of the contour will be distorted. Next, any unvoiced regions are filled in by using linear interpolation. Finally, 7 point smoothing is used to remove any "sharp edges" caused by the interpolation process.

3.2. RFC labeller

The RFC *element location* process tries to locate rise and fall elements from the smoothed F_0 contour. As intonational phenomena (such as pitch accents) are typically 100ms or longer, it was decided to re-sample the 5ms specified contour into 50ms frames so as to reduce the amount of data to be analysed using the location process.

The location algorithm is based on the simple principle that the F_0 contour in rise and fall elements changes more quickly than in other parts. Two trainable thresholds (the "gradient thresholds") are defined which are used by the system to decide whether a given frame is within a rise or fall element. Each frame is compared with the previous frame, and if there is a rise, and this rise is above the rise gradient threshold, this frame is labelled as a rise element. Likewise with fall elements. After this process is completed, adjacent frames marked with the same element are grouped together into sections. All other parts of the contour remained unlabelled at this stage.

This basic algorithm works quite well, but often errors occur due to the remaining obstruent influence. The effect of the obstruents is to introduce spurious rise and fall sections in the middle of unlabelled sections, or to split a single labelled section into two smaller sections separated by an unlabelled section. These errors were largely eliminated by introduction of a "deletion" process whereby sections that were below a certain length were deleted, and the legitimate sections on either side joined to make a single section. Two trainable thresholds (the "deletion thresholds") were defined that represented the minimum duration that a rise and fall element could have and the smallest gap of unlabelled contour that could exist between two labelled sections of the same type.

The element location algorithm only gives a rough (accurate to 50ms) indication of where the rise and fall elements occur. The *element boundary refinement process* decides where the precise boundaries of the rise and fall elements should be marked. A simple technique is used whereby a search region is defined using the rough boundaries produced by the location process. These search boundaries typically extend 150ms or so around the marked boundary. Within this region, every possible size of element is synthesized, and the one with the closest Euclidean distance to the contour is chosen as being the best fit. The element boundaries are then redefined to be where start and end of the best-fit element are. Any unlabelled sections are now labelled as connection elements.

3.3. HLCB Labeller

The HLCB labeller takes the output from the RFC labeller and produces a description of the tune in terms of **H, L, C** and **B** elements using the grammar defined in table 1.

Table 1 does not give concrete definitions of notions such as "starts late in syllable", "starts early in syllable" as the theory underlying the model has not yet been sufficiently developed to give formal definitions for these concepts. However, for practical purposes it is possible to define some thresholds which can be used to subclassify the accent types.

At this stage it is necessary to use information other than the F_0 contour alone as timing information is needed. Thus a segmental transcription is used to distinguish accent types. A "vowel-onset-to-fall" distance is defined which is the distance from the start of the vowel of the accented syllable to the start of the fall section. From studying the data described below, it was found that the fall in $\mathbf{H}_d$ accents typically occurred about 60ms after the vowel onset, whereas the fall of $\mathbf{L}_a$ accents typically occurred about 140ms *before* the vowel onset. Hence a rule was formulated such that fall elements occurring before the vowel onset are classed **L** and any after are classed **H**. Similarly, $\mathbf{H}_l$ accents had falls occurring about 100ms after the vowel onset. By using measures such as these, we can resolve the remaining ambiguity in mapping from RFC to HLCB descriptions. Although the timing rules proved very successful at distinguishing accent types, these rules are very ad-hoc, and a more thorough analysis of the data would be required before robust timing relationships were discovered.

The output of the HLCB system is a list of phonological elements, each with a basic type (**H**, **L** etc), a feature description ("late", "downstep" etc) and a position in ms which can be aligned with a segmental transcription to ascertain which syllable the element is associated with.

4. DATA, TRAINING, EXPERIMENTS AND RESULTS

4.1. Data

Two databases were used to test the system. When hand labelled, data set A contained 164 pitch accents and 136 intonation phrases in 64 utterances. Data Set B, from a different speaker, contained 301 pitch accents and 156 phrases in 45 utterances. F_0 contours were measured using both a laryngograph and a waveform F_0-tracking algorithm. Data Set A contained mostly voiced obstruent-free speech and designed so as to cover a wide variety of intonational tune types. Database B was much more spontaneous in nature, had no controls of segmental environment and was spoken by a speaker with no specialist knowledge of intonation.

4.2. Assessment

It is inherently difficult to assess systems such as this as we have no completely satisfactory method of determining how well the system is performing; the best we can do is to compare the transcriptions that the system produces with transcriptions from an expert human labeller. The trouble with such an approach is that we cannot guarantee that the transcriptions produced by the human are in any way reliable, as human intuition and arbitrary labelling criteria may have been used in the labelling process.

A sophisticated method of comparing and assessing transcriptions is given in Taylor [5], but here we can demonstrate the performance of the system with a simple measure of how many RFC and HLCB accents were correctly identified.

4.3. Training Method

As described above, the system is based on a simple set of thresholds. By systematically varying these thresholds, and measuring the error scores, it was possible to train a set of thresholds for a particular speaker or set of data.

The principle behind the training method is to select a wide range of values for each threshold, and construct a matrix were there is one axis for each threshold and each entry in the matrix corresponds to a different set of thresholds. The labelling system is run several times, each time using a different set of thresholds from the matrix. The recognition score is recorded for each run, and the matrix entry giving the best performance is then registered as being the set of optimal thresholds.

The value of the rise gradient threshold is varied from 20Hz/second to 800 Hz/second, and the value of the rise deletion threshold is varied from 0.025 seconds to 0.525 seconds. The fall set of thresholds are varied in the same way.

At present the training method is only used for the RFC labeller. Due to the provisional nature of some of the methods used in the HLCB labeller, a training method has not yet been devised for this part of the system and the thresholds are set by hand.

4.4. Results of RFC labelling

The training module produced the thresholds shown in table 2. It is clear from this table that the rise and fall

A	Rise gradient	120Hz/s
A	Fall gradient	-120Hz/s
A	Rise delete	0.125s
A	Fall delete	0.125s
B	Rise gradient	120Hz/s
B	Fall gradient	-140Hz/s
B	Rise delete	0.125s
B	Fall delete	0.125s

Table 2. *Trained thresholds for data sets A and B*

Data set	Element	number of tokens	% correct
A	Rise	103	80
A	Fall	94	93
A	Connection	128	81
A	All	325	84
B	Rise	244	69
B	Fall	213	88
B	Connection	283	72
B	Connection	740	75

Table 3. *Results for data sets A and B for laryngograph F_0 contours*

thresholds are very similar, and it is also clear that the thresholds don't vary much between the two speakers. This should not be taken as an indication that the rise and fall element behaviour is the same, (for instance fall elements are typically 50% longer than rise elements), but simply as an indication that the labelling system is quite crude and does not need fine adjustment for different sets of data.

The recognition scores using the best sets of thresholds (on laryngograph F_0 contours) are give in table 3. Overall the system recognises 84% of the elements in data set A, and 75% of the elements in data set B.

Data set A contains mostly voiced, obstruent free speech, whereas database B had no imposed segmental restrictions and thus has many more obstruents and unvoiced regions. As the F_0 processing module does not eliminate all the obstruent influence, more obstruent influence is present in the smoothed contours of data set B, and it is this fact that accounts for most of the deviation in performance between the two sets of data.

When F_0 contours from waveform based F_0-tracking algorithms are used, the performance is about 10% worse for both sets of data.

4.5. Results of HLCB labelling

As the thresholds used for labelling in the HLCB system work very well, it is really only errors in the RFC labelling that will produce automatic HLCB transcriptions which are different from the hand versions. In data base A, 93% of **H** accents and all **L** were correctly identified, but sometimes the subclass was different, e.g. too many accents were marked as being downstepped due to missing rise elements. In data set B, 88% of **H** accents and again all **L** were correctly identified.

At present, the system makes no distinction between nuclear and non-nuclear accents. From an informal evaluation of the transcriptions, it is obvious that the system correctly identifies larger accents (both in F_0 amplitude and duration) better than smaller accents. As nuclear accents are commonly larger than pre-nuclear accents, the system thus recognizes nuclear accents better, and this may be significant for any higher level system that is to interpret the output of the HLCB labeller. Thus the system performs best on those accents which are most important.

5. CONCLUSION

It is worth noting that this system works in an unconstrained manner, in that it pre-supposes no prosodic knowledge of the utterance it is analysing. In this way it differs from labelling systems which are given a transcription before hand and merely have to align that transcription with the utterance; and recognition systems, which have a grammar which gives makes the system select from a small number of possibilities at any given moment. Viewed in this way, the results are very promising. However, it is clear that much improvement can be made. The F_0 processor could be improved so as to eliminate more obstruent influence, perhaps by giving it explicit information of the segmental environment. The RFC labeller uses very simple techniques and its is possible that a much more sophisticated statistical recogniser could perform better and more work needs to be carried out in defining and training the thresholds in the HLCB labeller.

REFERENCES

[1] W. N. Campbell. Automatic detection of prosodic boundaries in speech. *Speech communication*, 1993. (forthcoming).

[2] Jim L. Hieronymus. Use of acoustic sentence level and lexical stress in hsmm speech recognition. In *International Conference on Speech and Signal Processing*. IEEE, 1992.

[3] J. D. O'Connor and G. F. Arnold. *Intonation of Colloquial English*. Longman, 2 edition, 1973.

[4] Janet B. Pierrehumbert. *The Phonology and Phonetics of English Intonation*. PhD thesis, MIT, 1980. Published by Indiana University Linguistics Club.

[5] Paul A. Taylor. *A Phonetic Model of English Intonation*. PhD thesis, University of Edinburgh, 1992.

[6] Paul A. Taylor. Synthesizing intonation using the rise/fall/connection model. In *Proc. ESCA Workshop on Prosody, Lund, Sweden*, 1993.

[7] A. Waibel. *Prosody in Speech Recognition*. PhD thesis, C.M.U., 1986.

[8] Colin W. Wightman, Stefanie Shattuck-Hufnagel, Mari Ostendorf, and Patti J. Price. Segmental durations in the vacinity of prosodic phrase boundaries. *Journal of the Acoustical Society of America*, 91:1707–1717, 1992.