DETECTION OF PHRASE BOUNDARIES IN JAPANESE BY LOW-PASS FILTERING OF FUNDAMENTAL FREQUENCY CONTOURS

Atsuhiro Sakurai and Keikichi Hirose
atsuhirou@gavo.t.u-tokyo.ac.jp hirose@gavo.t.u-tokyo.ac.jp

Dept. of Information and Communication Engineering, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113 Japan

ABSTRACT

Major syntactic boundaries are often accompanied by a rise in the phrase component of the fundamental frequency ($F_0$) contour. Detecting such rises, therefore, can be significantly helpful to the speech recognition process. We developed a method to detect syntactic boundaries with phrase-component rise (henceforth, phrase boundaries), based on the compression of the accent component of the $F_0$ contour (in logarithmic scale), using a low-pass filter. In this method, $F_0$ contours are viewed as signals in the time domain, which can be roughly separated into phrase and accent components due to their different frequency contents. Phrase boundaries are detected whenever a significant rise occurs in the derivative of the filtered $F_0$ contour. (The concepts of phrase and accent components can be found in [1]).

The method managed to detect about 77% of manually detectable phrase boundaries, though with a relatively high insertion rate. The insertion rate can be reduced by using the partial AbS method, proposed by the authors [7].

1. INTRODUCTION

We developed a method to detect phrase boundaries by compressing the accent component of the $F_0$ contour, using a low-pass filter. In this process, the derivative of the filtered $F_0$ contour is used to detect phrase boundaries.

There is, however, a mismatch between temporal events detected using information contained in the filtered $F_0$ contour and the corresponding phrase boundaries in the speech waveform due to the physical phrasing mechanism and the filtering process. (This issue has not been considered in a similar attempt by Ström [2].)

The mismatch caused by the phrasing mechanism can be compensated, to an acceptable degree, by means of a constant bias level. As to the mismatch caused by the filtering process, we showed that the relation between the derivative peak corresponding to a phrase boundary and the time lag of a temporal event (to be explained in more detail later) with respect to the boundary is approximately linear, making it possible to derive a linear function relating those events and the actual boundaries. The function is used in the last stage of the present method to detect phrase boundaries. Experiments were realized using the method, which proved to be valid for detecting phrase boundaries with errors of up to 1 mora.

2. DESCRIPTION OF THE METHOD

The method consists basically of the following steps: pre-processing, event detection and boundary detection.

The pre-processing consists of filtering the $F_0$ contour using a simple low-pass filter and calculating the derivative of the filtered curve. The second step, event detection, consists of detecting temporal events that denote the occurrence of a phrase boundary. The last step, i.e., boundary detection, consists of calculating the position of the phrase boundary based on the detected event and the corresponding derivative peak.

2.1. Pre-Processing

First, the following pre-processing was realized on the speech material:

1. $F_0$ contour extraction and correction of extraction errors

The $F_0$ contours are extracted using an autocorrelation function with the frame length proportional to the time lag [3]. An automatic algorithm was developed to detect and eliminate extraction errors: the extracted samples of the $F_0$ contour (voiced samples) are taken 3 by 3 (by shifting the moving window sample at a time), and the slopes of the segments defined by these samples are calculated. When the slope exceeds a threshold, the sample that caused the abnormal deviation from the rest of the contour is viewed as unvoiced.

2. Straight-line interpolation of the $F_0$ contour in a logarithmic scale

The unvoiced segments of the $F_0$ contour are linearly interpolated, based on the values at the edges (the last frame before the unvoiced segment and the first voiced
frame after the unvoiced segment.) This process suffers
the influence of spurious oscillations due to micro-
prosody, but such influence is absorbed by the later low-
pass filtering.

3. Low-pass filtering of the interpolated \( F_0 \) contour

The \( F_0 \) contour is filtered using a simple Butterworth
low-pass filter. The Butterworth filter was selected due
to its relatively flat phase characteristic.

4. Calculation of the derivative of the filtered curve

The derivative of the curve at any point is obtained by
taking the slope of the minimum square error line across
the set of 3 points formed by the point in question and
the two adjoining ones.

2.2. Low-Pass Filtering of the \( F_0 \) Contour

In order to satisfactorily eliminate the accent component of
the \( F_0 \) contour, the cut-off frequency should be on the order
of 1 to 2 Hz, as in [2]. However, if the cut-off frequency were
set within this range, the phrase component would also be
greatly affected by the filtering process, making the process
of phrase boundary detection more difficult. Here, we opted
to select a higher cut-off frequency, leaving vestiges of accent
components on the \( F_0 \) contour. The characteristics of the
filter utilized in the experiments are shown in Table 1.

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Butterworth Low-Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passband Frequency [Hz]</td>
<td>2</td>
</tr>
<tr>
<td>Stopband Frequency [Hz]</td>
<td>7</td>
</tr>
<tr>
<td>Order</td>
<td>3</td>
</tr>
</tbody>
</table>

2.3. Event Detection

After the pre-processing, the next step is to detect the events
in the derivative of the filtered \( F_0 \) contour that denote the
occurrence of a phrase command. In principle, the derivative
of an \( F_0 \) contour with no accent component would be a con-
stantly negative curve, becoming positive only at the verge
of a new phrase command. Therefore, some of the events
that could be used as indicators of new phrase components
in the \( F_0 \) contour are:

1. Negative-to-positive transitions of the derivative (zero-
crossings);
2. Another threshold involving the value of the derivative
curve;
3. A threshold involving the average of the derivative over
a moving window instead of the instantaneous value.

In order to find the most suitable event for the purpose of our
method, we performed experiments using model-generated

\[ F_0 \] contours containing an inner phrase boundary (from now
on, we will refer to a phrase boundary not located at the begin-
ing of the sentence as an inner phrase boundary), varying
the magnitude of the phrase command and the amplitude of
the subsequent accent command.

We generated \( F_0 \) contours for the following utterance:

- “sochirano kohusaiiga ini / roNbuNo tookoshitaito
  omonNdesuga.”
  (I'd like to submit a paper to this international conference.),

where the slash ‘/’ represents the syllabic position of the
phrase boundary in question.

The \( F_0 \) contours for the utterance above were generated us-
ing rules for speech synthesis [5], and both the magnitude of
the phrase command and the amplitude of the accent com-
mand were varied in the interval [0.1;0.6].

Figure 1 illustrates the case when the magnitude of the phrase
command (the phrase command located in the middle of the
sentence) and the amplitude of the subsequent accent com-
mand are 0.35. The generated curve is shown in (a),
and the \( F_0 \) model commands, in (b). The vertical line in (a)
represents the segmental position of the phrase boundary.
It should be noted that no other command besides the phrase
command in question and the following accent command was
varied.

For each case, events (1), (2) and (3) were detected and plot-
ted against the derivative peak corresponding to the bound-
dary in question. The value of the derivative peak is searched
within the interval beginning 200 ms before the event, ending
400 ms after. From the results, it was noted that event (3) is
the most suitable to detect the position of the phrase com-
mand, since the relation between event (3) and the derivative
peak was almost linear within the observed range of varia-
tion of the magnitudes of the phrase command. Moreover,
the variation of the derivative peak due to the variation of
the amplitude of the accent command, when plotted against
event (3), showed a similar behavior as when the magnitude of the phrase command was varied, which means that the influence of accent components can be easily handled using the same approach as for phrase components. The obtained curve can be seen in Figure 2.

In this method, therefore, phrase boundaries are determined using event (3) (when the average of the derivative of the filtered $F_0$ contour over a moving window exceeds a threshold) and the derivative peak at the boundary in question.

Using the average over a moving window instead of the instantaneous value seems logical in the present method. Since accent components are not eliminated completely, local rises corresponding to onsets of accent commands can result in negative-to-positive transitions of the derivative curve even when no phrase command is present. By using an averaging process, though, the algorithm becomes sensitive only to significant rises of the $F_0$ contour. The sensitivity of the search process can be controlled by the width of the moving window: a narrow window results in a highly sensitive algorithm, and a wide window makes the trigger occur only at long and substantial rises of the derivative. Here, after realizing the filtering and derivation process on a set of 25 $F_0$ contours (the same $F_0$ contours will be later utilized in evaluation experiments), and based on various observed values of widths of derivative peaks, we adopted a window width of 300 ms.

As to the threshold, a phrase boundary is detected whenever the mean value of the derivative peak in the window exceeds $7\times10^{-3}s^{-1}$.

Once the event is detected, the relation between the instant of occurrence of the event (EVENT TIME, from now on) and the actual position of the phrase boundary should be investigated.

From Figure 2, we derived a linear relationship between EVENT TIME and DERIVATIVE PEAK, allowing us to find the deviation of EVENT TIME from the actual phrase boundary.

$$B = ET + 4.77\times DP - 5.64\times10^{-2}[sec],$$

where $B$ represents BOUNDARY, $ET$ is EVENT TIME and $DP$ is DERIVATIVE PEAK.

3. EXPERIMENTAL RESULTS

A set of experiments were realized using $F_0$ contours extracted from the ATR Continuous Speech Database [4].

We selected 25 continuous speech samples, extracted their $F_0$ contours and determined the phrase commands of their underlying models using the AbS analysis [1]. Here, we did not deal with phrase boundaries occurring after pauses longer than 500 ms (respiratory pauses), which can be more easily detected using the temporal information of the pause.

Next, we applied the proposed method of automatic detection of phrase boundaries to the same set of $F_0$ contours in order to compare the results. The results of the experiments are summarized in Table 2.

<table>
<thead>
<tr>
<th>Inner Phrase Boundaries</th>
<th>56</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detected</td>
<td>43</td>
</tr>
<tr>
<td>Insertions</td>
<td>14</td>
</tr>
</tbody>
</table>

From the table, it can be seen that the method was able to find approximately 77% of the manually detectable phrase boundaries, with an insertion rate comparable to the deletion rate.

In the next set of experiments, we evaluated the method as to the deviation of the detected boundaries with respect to the actual boundaries in the speech waveform. We selected, among the phrase boundaries used in the previous experiments, 40 phrase boundaries that actually correspond to major syntactic boundaries, and determined the deviation of the detected boundaries with respect to their correct positions, in terms of number of morae. The results are described in Table 3.

<table>
<thead>
<tr>
<th>Inner Phrase Boundaries</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detected</td>
<td>14</td>
</tr>
<tr>
<td>1-mora Deviation</td>
<td>18</td>
</tr>
<tr>
<td>2-mora Deviation</td>
<td>34</td>
</tr>
</tbody>
</table>

Figure 3 shows an example of phrase boundary detection using the proposed method. The content of the utterance is
"daimokutoshitewa kanari kouchaNna kotoga kareteraruNdesuga yosurunyakudedeNwadesuka soreni laNnuni gijutsudeshitaranaNndemo yoroshitountaishakude yoroshiwake desune."  
(As to the title, they cover a wide variety of topics, but in short, if the theme has any relation with automatic interpreting telephone, or any related technological issues, then it will be fine, won't it?)

**Figure 3:** Phrase boundary detection (excluding phrases following long pauses) (a): waveform; (b): correct phrase boundaries; (c): detected phrase boundaries; (d): filtered $F_0$ contour; (e): derivative contour

4. CONSIDERATIONS

Other methods exist to detect boundaries using the $F_0$ contour [6], apparently with smaller insertion rates. However, these methods do not make any differentiation between phrase boundaries and non-phraseal boundaries (boundaries that are not associated with phrase commands), and thus their insertion rates do not reflect the same quantities treated in this work. Here, the number of false alarms represented by the insertion rate includes all the non-phraseal boundaries (due to accent commands) that are wrongly regarded as phrase boundaries.

In the present method, the experiments showed that eliminating insertions is an important issue that remains to be solved. (In the example shown in Figure 3, for instance, there is an insertion at 1.63 sec.) It was also noted that most of the insertions occurred when two phrase boundaries were detected close to each other (within an interval of less than 1.0 sec.)

The main reason for the insertion errors is the fact that the accent components are not completely compressed by the low-pass filter (Butterworth filter with a cut-off frequency of 7 Hz for the current experiment). Lowering the cut-off frequency would reduce insertion errors, but with the serious expense of lowering the detection rate. In order to deal with this problem, it is possible to use the "partial analysis-by-synthesis" method, formerly developed by the authors [7], by setting various recognition hypotheses with and without a phrase command. The candidate yielding the smallest ABs error could be assigned a higher probability due to its prosodic features.

5. CONCLUSION

The method of phrase boundary detection developed here, based on low-pass filtering of $F_0$ contours, proved valid to determine visually detectable phrase boundaries, with the detection rate of around 77%. The insertion rate can be further decreased by using the partial ABs method.

The method can be used in combination with an existent algorithm that automatically detects accent boundaries from the $F_0$ contour, based on information obtained from its derivative, contributing to a possible solution to the inverse problem of finding the $F_0$ model parameters from the $F_0$ contour.

6. REFERENCES