Improving phonetic alignment by handling secondary sequence structures
Johann-Mattis List

To cite this version:
Johann-Mattis List. Improving phonetic alignment by handling secondary sequence structures. Computational approaches to the study of dialectal and typological variation, organized as part of the ESSLLI 2012, Aug 2012, Opole, Poland. hprints-00742679v1

HAL Id: hprints-00742679
https://hal-hprints.archives-ouvertes.fr/hprints-00742679v1
Submitted on 16 Oct 2012 (v1), last revised 17 Oct 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Improving Phonetic Alignment by Handling Secondary Sequence Structures

Johann-Mattis List
Heinrich Heine University Düsseldorf
listm@phil.uni-duesseldorf.de
June, 2012

Primary and Secondary Sequence Structures

In traditional alignment analyses, sequences are only compared with regard to their primary structure. Here, the term primary structure refers to the order of segments, whereby segments are understood as the smallest units of a sequence which directly correspond to the characters of the alphabet from which the sequence is drawn. Apart from the primary structure, sequences can, however, also have a secondary structure. Apart from segmentizing sequences into their primary units, one can further segmentize them into larger units of subsequences consisting of one or more primary segments. A secondary segmentation which is very common in linguistics is, e.g., the segmentation of words into syllables apart from the primary segmentation of words into phonemes.

The traditional alignment modes such as global, local, or semiglobal alignment (cf. the overview in Durbin et al. 2002) align sequences only with respect to their primary structure. Thus, given the sequence "THE CATFISH HUNTS" and "THE CAT FISHES", they all yield an alignment in which the subsequence "CATFISH" of the first sequence is matched with the subsequence "CAT FISH" of the second sequence (see Table 1). In contrast to these primary alignments, a secondary alignment displays the similarities of sequences with regard to both their primary and their secondary structure, aligning letters which belong to the same word in one sequence only with those letters in the other sequence which also belong to a single word (see Table 1).

<table>
<thead>
<tr>
<th>Mode</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Alignment</td>
<td>T H E  C A T - F I S H H U N T S</td>
</tr>
<tr>
<td></td>
<td>T H E  C A T - F I S H - E - - - S</td>
</tr>
<tr>
<td>Secondary Alignment</td>
<td>T H E  C A T - F I S H H U N T - S</td>
</tr>
<tr>
<td></td>
<td>T H E  C A T - - - F I S H E S</td>
</tr>
</tbody>
</table>

Table 1: Primary vs. Secondary Alignment

Secondary alignment analyses are especially useful when dealing with South-East Asian tonal languages like Chinese, since in these languages the morphemes are almost exclusively monosyllabic, while the words usually are not. In contrast to primary alignment analyses, secondary alignment analyses can keep track of the syllable boundaries and thus help to avoid matching the sounds of one syllable in one word.
with sounds in two syllables of the other. For example, a traditional alignment analysis of Haikou Chinese
[zi] “sun” with Beijing Chinese [zi] “sun” usually wrongly matches the dental plosives
of both words, ignoring that one word has only one morpheme, while the other one has two, as shown in
Table 2.¹

<table>
<thead>
<tr>
<th>Primary Alignment</th>
<th>Secondary Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haikou</td>
<td>zi - t - ³</td>
</tr>
<tr>
<td>Beijing</td>
<td>zi ⁴ t⁴ ou ¹</td>
</tr>
</tbody>
</table>

Table 2: Secondary Alignment of Tonal Languages

The Basic Algorithm for Pairwise Alignment

The basic algorithm for the computation of an optimal alignment of two sequences was independently
developed by different scholars from different scientific disciplines. In biological applications this algorithm
is usually called Needleman-Wunsch algorithm, named after S. B. Needleman and C. D. Wunsch who
published their algorithm as a solution for the sequence alignment problem in biology in 1970 (Needleman
and Wunsch 1970). In general applications of computer science it is also common to refer to the algorithm as
Wagner-Fischer algorithm, named after R. A. Wagner and M. J. Fischer’s algorithm for the computation
of the edit distance between two strings (Wagner and Fischer 1974). Both algorithms and their various
extensions belong to the family of dynamic programming algorithms (DPA, cf. Eddy 2004, Gusfield 1997:
217f). The main idea of dynamic programming is to find an approach for the solution of complicated
problems ‘that essentially works the problem backwards’ (Rosenberg 2009b: 4). Thus, instead of checking
all possible alignments between two sequences and looking for the best one in order to find an optimal
alignment, an alignment is built up ‘using previous solutions for optimal alignments of smaller subsequences’
(Durbin et al. 2002: 19).

The core part of the Needleman-Wunsch algorithm, the creation of the alignment matrix, in the tradi-
tional global mode is given in pseudo-code in Algorithm 1. Since the Needleman-Wunsch algorithm is
well-described in the literature (see, e.g., Durbin et al. 2002, Gusfield 1997, Kondrak 2002), I will not give
a detailed description in this context.

A New Algorithm for Pairwise Secondary Alignment

In order to make the traditional alignment modes sensitive for secondary sequences structures, only some
slight modifications of the basic algorithm for pairwise sequence alignment are needed. Assuming that the
secondary structure of sequences is marked by some boundary marker r, which is introduced as a separator
between all secondary segments (such as a whitespace in sentences or tone letters in phonetic sequences
drawn from tonal languages), two restrictions have to be added to the main loop: (1) one restriction which
prohibits the matching of the boundary marker with all other segments, and (2) one restriction which pro-
hibits the matching of the boundary marker with a gap which is introduced inside a secondary segment.
These restrictions can be implemented by simply setting the relevant scores to −∞. In order to check
whether the first condition holds for a certain cell in the matrix, the scoring function can be modified in
such a way that it yields 0 for the matching of boundary markers, and −∞ for the matching of the boundary
marker with any other character. The check for the second condition can be implemented with help of an
if-statement. The pseudo-code for the modification of the main loop is given in Algorithm 2. As input

¹Data for the Chinese dialects is taken from (Hóu 2004).
Algorithm 1: GlobalAlignMatrix\((x, y, g, \text{score})\)

/* matrix construction */
\[ M \leftarrow \text{matrix}(\text{length}(x) + 1, \text{length}(y) + 1) \]
\[ M[0][0] \leftarrow 0 \]

/* matrix initialization */
for \(i \leftarrow 1\) to \(\text{length}(x)\)
do \[ M[i][0] \leftarrow M[i][0] + g \]
for \(i \leftarrow 1\) to \(\text{length}(y)\)
\[ M[0][i] \leftarrow M[0][i] + g \]

/* main loop */
for \(i \leftarrow 1\) to \(\text{length}(x)\)
\[ \begin{align*}
   \text{for } j \leftarrow 1 \text{ to } \text{length}(y) \\
   \text{do } M[i][j] & \leftarrow \max \left\{ M[i-1][j-1] + \text{score}(x_{i-1}, y_{j-1}), \\
                     M[i-1][j] + g, \\
                     M[i][j-1] + g \right\}
\end{align*} \]

parameters, there are the two sequences \(x\) and \(y\), the gap penalty \(g\), the boundary marker \(r\), and the scoring function \text{score}().

Evaluation

In order to test the performance of the modified algorithm compared to the traditional algorithm for phonetic alignment, I carried out global alignment analyses on a testset (see Online Material 1) consisting of 1 089 manually aligned sequence pairs drawn from Chinese (Hóu 2004) and Bai dialects (Allen 2007, Wang 2006). Since both Chinese and Bai are tonal languages with a monosyllabic morpheme structure, alignment analyses which are sensitive to syllable boundaries should certainly improve traditional alignment analyses. Data drawn from tonal languages therefore offers a good test case for the new approach.

The new approach for secondary alignment was introduced in the most recent version of the SCA (Sound-Class-Based Phonetic Alignment) method (List 2012) which is implemented as part of the LingPy library (Version 1.0). \(^2\) Using the default settings of SCA two different analyses were carried out, one primary analysis and one secondary analysis, where the basic algorithm of SCA was modified according to Algorithm 2, and tone letters were defined as boundary markers. In order to test how well both methods performed in comparison with the gold standard, two evaluation scores, the Column score (CS) and the

\(^2\)Online available under http://lingulist.de/lingpy/.
Algorithm 2: SecondaryAlignmentMatrix($x$, $y$, $g$, $r$, $\text{score}$)

/* main loop */
for $i$ ← 1 to length($x$) do
  for $j$ ← 1 to length($y$) do
    /* score() checks for restriction 1 */
    $M[i][j] \leftarrow \max$
    \[
    M[i - 1][j - 1] + \text{score}(x_{i-1}, y_{j-1})
    \]
    /* check for restriction 2 */
    if $x_{i-1} = r$ and $y_{j-1} \neq r$ and $j \neq \text{length}(y)$
      then $-\infty$
    else $M[i - 1][j] + g$
    if $y_{j-1} = r$ and $x_{i-1} \neq r$ and $i \neq \text{length}(x)$
      then $-\infty$
    else $M[i][j - 1] + g$

Sum of Pairs score (SPS) were computed. The column score is defined as:

\[ CS = 100 \cdot 2 \cdot \frac{|C_t \cap C_r|}{|C_r| + |C_t|} \]  \hspace{1cm} (1)

where $C_t$ is the set of columns in the test alignment and $C_r$ is the set of columns in the reference alignment (Rosenberg and Ogden 2009). The SPS is defined as:

\[ SPS = 100 \cdot 2 \cdot \frac{|P_t \cap P_r|}{|P_r| + |P_t|} \]  \hspace{1cm} (2)

where $P_t$ is the set of all aligned residue pairs in the test alignment and $P_r$ is the set of all aligned residue pairs in the reference alignment (ibid.). In addition, the proportion of perfectly aligned sequences (PAS) is also reported.

<table>
<thead>
<tr>
<th>Score</th>
<th>Primary</th>
<th>Secondary</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAS</td>
<td>83.47</td>
<td>88.89</td>
</tr>
<tr>
<td>CS</td>
<td>88.54</td>
<td>92.70</td>
</tr>
<tr>
<td>SPS</td>
<td>92.78</td>
<td>95.52</td>
</tr>
</tbody>
</table>

Table 3: Results of the Evaluation

Results and Conclusion

As can be seen from the results given in Table 3 (see also the Supporting Online Material), the modified algorithm which is sensitive to secondary sequence structures shows a great improvement compared to the
traditional algorithm which aligns sequences only with respect to their primary structure. The improvement is significant with $p < 0.01$ using the Wilcoxon signed rank test as suggested by (Notredame2000). The algorithm for secondary alignment proves very useful for the alignment of tonal languages, yet it may also be employed for the analysis of other kinds of sequential data and, e.g., help to carry out phonetic alignment analyses of whole sentences.

References


Online Material

The Supporting Online Material for this study can be downloaded from http://lingulist.de/supp/secondary.zip.