Improving phonetic alignment by handling secondary sequence structures

Johann-Mattis List

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 *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).

Mode	Alignment
Primary Alignment	T H E _ C A T – F I S H _ H U N T S T H E _ C A T _ F I S H – E – – – S
Secondary Alignment	T H E _ C A T F I S H _ H U N T – S T H E _ C A T – – – – _ F I S H E S

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

Hinrichs, Erhard and Jäger, Gerhard: "Computational approaches to the study of dialectal and typological variation". Workshop organized as part of the ESSLLI 2012.

with sounds in two syllables of the other. For example, a traditional alignment analysis of Haikou Chinese 日 [zit^3] "sun" with Beijing Chinese 日頭 [ʐʅ^{51}t^{h}ou^1] "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]

Primary Alignment						
Haikou	z	i	-	t	-	^{3}
Beijing	ʐ	ʅ	^{51}	t^h	ou	^{1}

Secondary Alignment							
Haikou	z	i	t	^{3}	-	-	-
Beijing	ʐ	ʅ	-	^{51}	t^h	ou	^{1}

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 build 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 traditional 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 prohibits 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 $-\infty$. 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 $-\infty$ 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

[1] Data for the Chinese dialects is taken from (Hóu 2004).

Algorithm 1: GlobalAlignMatrix(x, y, g, score)

/* matrix construction */

$M \leftarrow$ matrix(length(x) + 1, length(y) + 1)
$M[0][0] \leftarrow 0$

/* matrix initialization */

for $i \leftarrow 1$ **to** length(x)
  do $M[i][0] \leftarrow M[i][0] + g$
for $i \leftarrow 1$ **to** length(y)
  do $M[0][i] \leftarrow M[0][i] + g$

/* main loop */

$$\textbf{for } i \leftarrow 1 \textbf{ to } \text{length}(x) \quad \textbf{do} \begin{cases} \textbf{for } j \leftarrow 1 \textbf{ to } \text{length}(y) \\ \quad \textbf{do } M[i][j] \leftarrow \max \begin{cases} M[i-1][j-1] + \text{score}(x_{i-1}, y_{j-1}) \\ M[i-1][j] + g \\ M[i][j-1] + g \end{cases} \end{cases}$$

parameters, there are the two sequences x and y, the gap penalty g, the boundary marker r, and the scoring function 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, score)

```
/* main loop */
for i ← 1 to length(x)
    do { for j ← 1 to length(y)
         do { M[i][j] ← max { /* score() checks for restriction 1 */
                              M[i − 1][j − 1] + score(x_{i−1}, y_{j−1})
                              /* check for restriction 2 */
                              if x_{i−1} = r and y_{j−1} ≠ r and j ≠ length(y)
                                then −∞
                                else M[i − 1][j] + g
                              if y_{j−1} = r and x_{i−1} ≠ r and i ≠ length(x)
                                then −∞
                                else M[i][j − 1] + g
```

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

$$CS = 100 \cdot 2 \cdot \frac{|C_t \cap C_r|}{|C_r| + |C_t|}., \tag{1}$$

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:

$$SPS = 100 \cdot 2 \cdot \frac{|P_t \cap P_r|}{|P_r| + |P_t|}, \tag{2}$$

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.

Score	**Primary**	**Secondary**
PAS	83.47	88.89
CS	88.54	92.70
SPS	92.78	95.52

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

Allen, B. (2007). *Bai Dialect Survey*. SIL International. URL: `http://www.sil.org/silesr/2007/silesr2007-012.pdf`.

Durbin, R., S. R. Eddy, A. Krogh, and G. Mitchinson (2002). *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*. 7th ed. Cambridge: Cambridge University Press.

Eddy, S. R. (2004). "Where did the BLOSUM62 alignment score matrix come from?" In: *Nature Biotechnology* 22.8, 1035–1036.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences*. Cambridge: Cambridge University Press.

Hóu Jīng 侯精, ed. (2004). *Xiàndài Hànyǔ fāngyán yīnkù* 现代汉语方言音库 [Phonological database of Chinese dialects]. CD-ROM. Shanghai.

Kondrak, G. (2002). "Algorithms for language reconstruction". Dissertation. Toronto: University of Toronto. url: `http://webdocs.cs.ualberta.ca/~kondrak/papers/thesis.pdf`.

List, J.-M. (2012). "SCA. Phonetic alignment based on sound classes". In: *New directions in logic, language, and computation*. Ed. by M. Slavkovik and D. Lassiter. LNCS 7415. Berlin and Heidelberg: Springer, 32–51.

Needleman, S. B. and C. D. Wunsch (1970). "A gene method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of Molecular Biology* 48, 443–453.

Rosenberg, M. S., ed. (2009a). Berkeley, Los Angeles, and London: University of California Press.

– (2009b). "Sequence alignment. Concepts and history". In: *Sequence alignment. Methods, models, concepts, and strategies*. Ed. by M. S. Rosenberg. Berkeley, Los Angeles, and London: University of California Press, 1–22.

Rosenberg, M. S. and T. H. Ogden (2009). "Simulation approaches to evaluating alignment error and methods for comparing alternate alignments". In: *Sequence alignment. Methods, models, concepts, and strategies*. Ed. by M. S. Rosenberg. Berkeley, Los Angeles, and London: University of California Press, 179–207.

Wagner, R. A. and M. J. Fischer (1974). "The string-to-string correction problem". In: *Journal of the Association for Computing Machinery* 21.1, 168–173.

Wang, F. (2006). *Comparison of languages in contact. The distillation method and the case of Bai*. Taipei: Institute of Linguistics Academia Sinica.

Online Material

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