PHONEME-TO-PHONEME ALIGNMENT AND CONVERSION

Uwe D. Reichel, Raphael Winkelmann

Institute of Phonetics and Speech Processing, University of Munich
reichelu@phonetik.uni-muenchen.de

Abstract: This paper deals with new methods for phoneme-to-phoneme (P2P) alignment and conversion. Alignment is carried out by dynamic programming for Levenshtein distance calculation. Cost functions based on phoneme co-occurrence statistics and on distinctive feature vector distances accounting for connected speech processes are comparatively evaluated. Given the aligned data, decision trees for P2P conversion across word boundaries are trained and evaluated. Amongst others it turned out, that while accounting for assimilation processes improved alignment quality, these quality differences showed no impact on P2P conversion performance.

1 Introduction

P2P alignment serves to relate corresponding parts of parallel transcriptions. In fundamental linguistic research it can be used to infer phonological rules transforming canonic to spontaneous speech. In text-to-speech synthesis it serves in combination with P2P conversion to map the canonical transcription derived from text preprocessing onto a more natural transcription accounting for connected speech processes. Some approaches for proper name pronunciations [10] employ a cascade of grapheme-to-phoneme and subsequent correcting P2P conversion.

Alignment Alignment of two sequences a and b is generally treated as a task to transform a to b by a minimum amount of edit costs. This minimisation task can be solved algorithmically e.g. by a method proposed by [11]. In the simplest case the minimum edit cost corresponds to the minimum number of needed edit operations, the *Levenshtein distance*. In our study some cost functions related to typical edit operations and the affected symbols are introduced.

Conversion While alignment is generally learned in an unsupervised fashion, for P2P conversion supervised machine learning methods can be trained on the aligned data. Basically, the same methods can be used as for grapheme-to-phoneme conversion, as are for example probabilistic approaches like joint-sequence models [1] or classifiers like neural networks and decision trees [7]. While probabilistic converters attempt to find global solutions by maximising the overall conversion probability, the other mentioned models operate locally by converting a grapheme or phoneme with respect to its surrounding context to a target phoneme.

2 Alignment

2.1 General architecture

The models developped here have in common that they consider alignment as a dynamic programming problem as described in section 1. For all models the standard basic edit operations

substitution, insertion, and *deletion* are used. The models differ with respect to the costs assigned to the edit operations. While model *CoocA* employs a cost function based on phoneme co-occurrence probabilities, model *FeatA* assigns edit costs as suggested by phonological principles concerning phoneme representations, their functional loads, as well as connected speech processes.

Notation: In the subsequent sections the phoneme sequences v and w stand for the the canonic and the spontaneous speech transcriptions, respectively, $|v|$ for the length of v. c is a cost function, $c(v_i, w_j)$ denotes the cost to substitute the i-th symbol of the canonic transcription by the j-th symbol of the spontaneous speech transcription, $c(v_i, \_)$ is the cost to delete v_i, and $c(\_, w_j)$ is the cost to insert w_j.

2.2 *CoocA:* Statistic co-occurrence-based cost function

2.2.1 Substitutions

Substitution costs are defined as follows:

$$c(v_i, w_j) = \begin{cases} 0 & : \quad v_i == w_j \\ 1 - P(w_j|v_i) & : \quad \text{else.} \end{cases} \tag{1}$$

Zero substitutions are given a cost of zero, and substitution costs for unequal $< v_i, w_j >$-pairs are defined in terms of conditional probabilities, which are calculated by maximum likelihood estimation: $P(w_j|v_i) = \frac{\#(v_i, w_j)}{\#(v_i)}$. For the $< v_i, w_j >$ co-occurrence statistics counts are incremented by centering a triangular window of length 7 and area 1 on v_i. This window allows a weighted distribution of $< v_i, w_k >$ count increments for $k = j-3 \ldots j+3$. This procedure is very similar to the Pfitzinger aligner approach described in [6].

2.2.2 Deletions

The deletion cost of a phoneme v_i is defined as follows:

$$c(v_i, \_) \quad = \quad 1 - P(v_i \big| |v| > |w|). \tag{2}$$

To derive the underlying $< v_i, \_ >$ frequencies, transcription pairs containing a v which is longer than w trigger a uniform distribution of the total count increment $|v| - |w|$ over all v_i in v.

2.2.3 Insertions

The insertion cost of a phoneme w_j is defined as:

$$c(\_, w_j) \quad = \quad 1 - P(w_j \big| |v| < |w|). \tag{3}$$

As for the deletion costs, transcription pairs containing a v shorter than w, the count increment $|w| - |v|$ is distributed uniformly over all w_j in w.

2.3 *FeatA:* Similarity-based cost function

In the *FeatA* model costs are defined with respect to phonologic distances between phonemes which are expressed as *Hamming* distances between their distinctive feature vectors. Four variants of this model are introduced:

- **FeatA:** including feature weighting and assimilation
- **FeatA-a:** without assimilation
- **FeatA-aw:** without assimilation and feature weighting
- **FeatA-w:** without feature weighting

2.3.1 Distinctive Features and their weights

Vowels are described by the distinctive features *tongue height, tongue position*, and *lip rounding*, consonants by *mode, place*, and *voicing*. For /@/ the feature values are not specified which serves to faciliate vowel reduction and /@/-deletion (see below).

In *FeatA* and *FeatA-a* features are weighted by their *functional load FL*. In general, the FL of an opposition between two linguistic entities a and b (e.g. phonemes) is traditionally related to the number of contrasts this opposition is responsible for in a language L. The basic information theoretic definition adopted here was first introduced in [2]:

$$FL(a,b) \quad = \quad \frac{H(L) - H(L_{a=b})}{H(L)}. \tag{4}$$

$H(L)$ is the entropy of a language L. $L_{a=b}$ denotes a language lacking an opposition of a and b. $FL(a,b)$ thus stands for the relative amount of information loss resulting from such a merging.

In this study, we adopt this notion to quantify the importance of the distinctive features used for phoneme description. The functional load of a feature x is defined accordingly as:

$$FL(x) \quad = \quad \frac{H(L) - H(L_{-x})}{H(L)}, \tag{5}$$

L_{-x} assigning a language where the feature x is not used to distinguish between phonemes. Since our entropy calculation is based on word-level unigram probabilities, functional load is roughly related to the relative increase of the amount of canonic homophones in our training data resulting from the '$-x$'-induced phoneme merging.

FL is used to weight the contribution of each feature in distance calculation.

2.3.2 Substitutions

Substitution costs between phonemes v_i and w_j basically are defined as the Hamming distance d between their feature vectors f:

$$c(v_i, w_j) = d(f(v_i), f(w_j)) = \begin{cases} \frac{\sum_{n: fn(v_i) \neq fn(w_j)} 1}{|f(v_i)|} & : \textit{FeatA-aw, FeatA-w} \\ \frac{\sum_{n: fn(v_i) \neq fn(w_j)} fl_n}{|f(v_i)|} & : \textit{FeatA, FeatA-a}, \end{cases} \tag{6}$$

where n represents the positions at which $f(v_i)$ and $f(w_j)$ differ. Since vowel-consonant vector pairs contain incompatible features, they are assigned the substitution cost 1.

d is not entirely symmetric, since only unspecified values in $f(v_i)$ and not in $f(w_j)$ are taken into account for distance calculation. This asymmetry serves to capture vowel reductions.

Assimilation For models *FeatA* and *FeatA-w* equation 6 is expanded by following assignment:

$$d(f(v_i), f(w_j)) \quad = \quad d(fa(v_{i-1}, v_i, w_j), f(w_j)), \tag{7}$$

where *fa* is the vector most similar to $f(w_j)$ resulting from merging $f(v_{i-1})$ and $f(v_i)$. This procedure allows for modelling progressive assimilation of features of the preceeding segment. In case of $v_{i-1} = w_j$, assimilation is suppressed in order to model the articulatory constraint not to produce the same phoneme consecutively. It is not possible to model regressive assimilation of subsequent phoneme features since the dynamic programming algorithm used here lacks a look ahead mechanism.

2.3.3 Deletions

The deletion cost of v_i reflects the loss of syntagmatic contrast that would have been given in presence of v_i. In the models not accounting for assimilation (*FeatA-a, FeatA-aw*) this notion is defined by the FL-sum of all features of v_i normalised by the length of the feature vector. Combined with assimilation (*FeatA, FeatA-w*) the additional contrast is retrieved by summing the FLs of all features with values different from those of the preceeding phoneme v_{i-1}.

$$c(v_i, \_) = \begin{cases} \frac{\sum_n fl_n}{|v_i|} & : \quad \textit{FeatA-a} \\ \frac{\sum_n 1}{|v_i|} & : \quad \textit{FeatA-aw} \\ \frac{\sum_{n: f_n(v_i) \neq f_n(v_{i-1})} fl_n}{|v_i|} & : \quad \textit{FeatA} \\ \frac{\sum_{n: f_n(v_i) \neq f_n(v_{i-1})} 1}{|v_i|} & : \quad \textit{FeatA-w} \end{cases} \tag{8}$$

2.3.4 Insertions

In analogy to deletion, the cost of inserting w_j after v_i reflects the notion of newly introduced syntagmatic contrast, expressed again in model-dependent ways:

$$c(\_, w_j) = \begin{cases} \frac{\sum_n fl_n}{|w_j|} & : \quad \textit{FeatA-a} \\ \frac{\sum_n 1}{|w_j|} & : \quad \textit{FeatA-aw} \\ \frac{\sum_{n: f_n(w_j) \neq f_n(v_i)} fl_n}{|w_j|} & : \quad \textit{FeatA} \\ \frac{\sum_{n: f_n(w_j) \neq f_n(v_i)} 1}{|w_j|} & : \quad \textit{FeatA-w} \end{cases} \tag{9}$$

Note that in *FeatA-aw* insertion and deletion costs are simply 1.

3 P2P conversion

For P2P conversion C4.5 decision trees [5] are utilised. The mapping of the canonic phoneme v_i onto the connected-speech phone w_i depends on the following features:

- phoneme window $v_{i-2} \ldots v_{i+2}$
- word class: function or content word

- previous target phoneme w_{i-1} (which in testing and application is the preceeding tree output)

The word class feature is expected to account for the fact that more reductions take place in function words compared to content words. The recurrent architecture of the tree taking its previous output as an input feature is motivated by the need for phonotactic smoothness [8]. Since connected speech is to be captured, the phoneme window, as well as the previous target are not limited by word boundaries (except for those at turn boundaries).

4 Evaluation

4.1 Data and Method

For training and testing our aligners and converters we used the Kiel Corpus of Spontaneous speech [3], which provides manually aligned canonic and phonetic transcriptions of approximately 48000 word tokens.

To comparatively evaluate the aligners, for each alignment method a tenfold cross-validation was carried out. The training data served to estimate the probabilistic cost function for *CoocA* and the FL weights for *FeatA* and *FeatA-a*. In the test partititons only differing canonic and phonetic transcriptions were considered.

Subsequently, in order to examin the impact of alignment quality on P2P conversion, the tree classifier was trained and tested by tenfold cross-validation on five variants of the data produced by the five aligners.

For both alignment and P2P conversion word error rates (WER) and Mean normalised Levenshtein distance (MNLD) were measured. The MNLD is defined as in [8] as the average Levenshtein distance between model outputs and respective reference sequences normalised by the length of the reference. Costs for zero-substitutions were set to 0, all other editing costs to 1.

4.2 Alignment

Functional loads As can be seen in figure 1 the functional loads of the distinctive features differ significantly (Friedman test, $\chi^2_5 = 49.09$, $p < 0.001$). The consonant-related features mode *mod* and place of articulation *pla* carry a significantly higher load than the other features, which can be further divided into the vowel-related feature tongue height *hgt* and the significantly lower remaining features tongue position *pos*, lip rounding *rnd* and voicing *voi* (Dunnett post-hoc test, $\alpha = 0.05$).

Performance The left half of Figure 2 shows the word error rates of the compared alignment models on the training and the test data respectively after 10-fold cross validation.

Significant performance differences can be reported for the held-out test data (Friedman test, $\chi^2_4 = 37.84$, $p < 0.001$) showing that the phoneme similarity-based models accounting for assimilation, namely *FeatA* and *FeatA-w*, yield significantly better alignments than the others (Dunnett post-hoc test, $\alpha = 0.05$). Next to assimilation the weighting of the distinctive features leads to a slight but not significant further performance improvement.

A similar pattern arises when looking at mean normalised Levenshtein distances (see table 1), although a smaller number of significant differences can be found here.

No significant performance difference with respect to training against test data was observed for any of the alignment models, neither for WER nor for MNLD (Mann-Whitney tests, $p > 0.38$, mostly > 0.7).

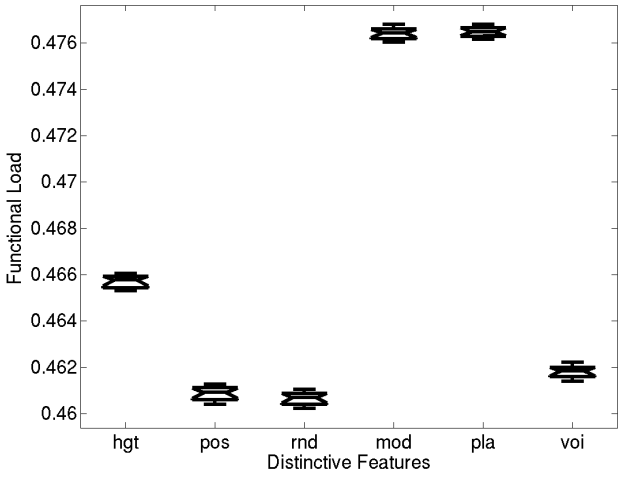

Figure 1 - Functional loads related to the utlised distinctive features after ten-fold cross validation. **Vowels:** *hgt*: tongue height, *pos*: tongue position, *rnd*: lip rounding, **Consonants:** *mod*: modus, *pla*: place of articulation, *voi*: voicing.

Table 1 - Mean normalised Levenshtein distances between aligner outputs and reference alignments and between P2P converter outputs and reference transcriptions depending on the aligner.

	Model	*CoocA*	*FeatA*	*FeatA-a*	*FeatA-aw*	*FeatA-w*
MNLD	Alignment	0.0335	0.0204	0.0236	0.0222	0.0222
	P2P	0.1030	0.1028	0.1034	0.1037	0.1049

4.3 Conversion

The right half of Figure 2 shows the word error rates of P2P conversion depending on the preceeding data alignment. No significant performance differences were found (Kruskal-Wallis test, $\chi^2_4 = 3.58,\ p = 0.47$). The same holds for the mean normalised Levenshtein distances shown in table 1.

5 Discussion

5.1 Alignment

Generalisation capability Since none of the alignment models performed significantly worse on the held-out data compared to the training data, it can be concluded that the methods introduced here are robust enough to cope with unseen data.

Due to its purely co-occurrence based cost-function *CoocA* does not explicitely utilise phonological knowledge and therefore can be applied also in other domains than P2P. The more phonologically motivated *FeatA* and its variants make usage of phonologic principles related to functional loads, feature vector distances, and assimilation processes of phonemes. Nevertheless, parts of the phonologic knowledge are not hard-coded but induced from the training data. A potential strength of the phonologic approaches may lie in comparably modest requirements concerning the amount of training data.

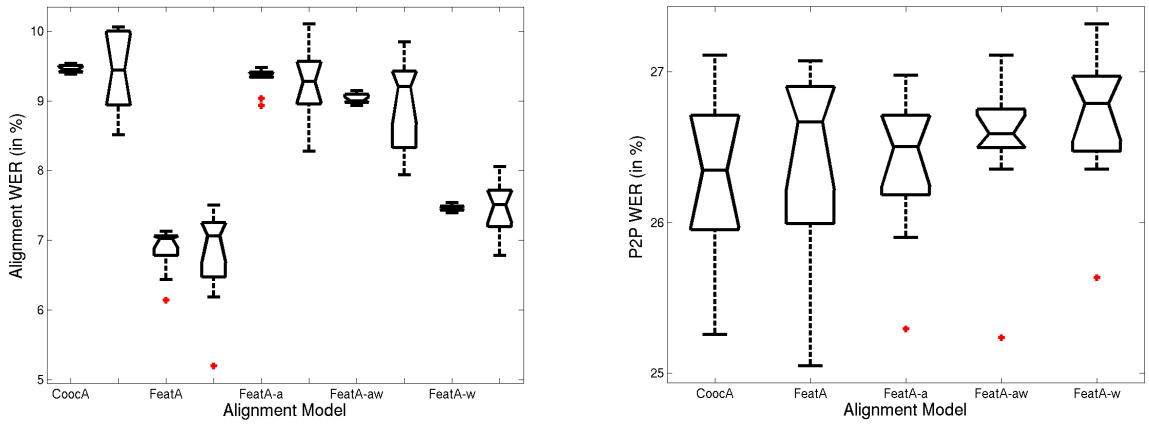

Figure 2 - **Left:** Alignment word error rates for all alignment models. **Right:** P2P conversion word error rates depending on the data alignment. **Alignment models:** *CoocA:* Co-occurrence-based, *FeatA:* Phoneme similarity-based including assimilation and feature weighting, *FeatA-a:* without assimilation, *FeatA-aw*: without assimilation and feature weighting, *FeatA-w:* without feature weighting.

Impact of phonologic knowledge Incorporating assimilation turned out to be valuable for P2P alignment resulting in significantly higher performances compared to all other models.

Functional load in contrast turned out to be of minor importance, since its integration led only to small and not significant improvements. This observation is in line with studies in other domains as sound change [9] reporting only little influence of FL. Perhaps a modified definition of this concept, e.g. by taking lexical context into account, could make the FL measure more valuable for these purposes.

Heuristics While for *CoocA* as well as for *FeatA* substitution costs are well motivated, insertion and deletion (*indel*) costs are assigned in a rather heuristic fashion. This shortcoming may weaken the general adequacy of the used metrics. While for example several heuristic methods are reviewed in [4], no more fundamental approaches to determine *indel* costs are known to the authors. Given this state of the art, at least alternative heuristics should be comparatively evaluated in future studies.

5.2 Phoneme-to-phoneme conversion

Alignment quality did not show any impact on P2P conversion accuracy, since significant alignment performance differences did not result in different P2P qualities. This finding suggests, that all employed alignment methods are equally adequate in providing P2P training data despite their quality differences.

In general P2P conversion yields rather modest performance rates which partly can be put into perspective by looking at following factors: First, as opposed to grapheme-phoneme conversion resulting in generally singular canonic transcriptions, P2P conversion has to face a much higher amount of target phoneme variability, which can be attributed to inter-speaker differences as well as intra-speaker variations triggered by higher linguistic factors like phonetic reduction processes in conveying *given* or *unimportant* as opposed to *new* or *important* information. These factors are not accounted for by the pool of low-level features used in this study.

Second, since the listener accepts (and also expects) pronunciation variability, the WER measure is certainly too crude to appropriately capture the P2P quality.

To cope with inter-speaker variability, P2P conversion could be restricted to model just a single speaker. Parts of the intra-speaker variability can be accounted for by providing the converter with higher linguistic semantics and discourse level features. Further, to consider also the variability accepted by the listener, next to the crude mathematical evaluation also perception experiments yielding comprehensibility and naturalness judgments are needed.

References

[1] BISANI, M. and H. NEY: *Joint-sequence models for grapheme-to-phoneme conversion*. Speech Communication, 50:434–451, 2008.

[2] HOCKET, C.: *The Quantification of Functional Load*. Word, 23:320–339, 1967.

[3] KOHLER, K.: *Labelled Data Bank of Spoken Standard German – The Kiel Corpus of Read/Spontaneous Speech*. In *Proc. ICSLP*, pp. 1938–1941, 1996.

[4] KONDRAK, G.: *Algorithms for Language Reconstruction*. PhD thesis, University of Toronto, 2002.

[5] QUINLAN, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.

[6] REICHEL, U. and H. PFITZINGER: *Text Preprocessing for Speech Synthesis*. In *Proc. TC-Star Speech to Speech Translation Workshop*, pp. 207–212, Barcelona, Spain, 2006.

[7] REICHEL, U., H. PFITZINGER and H. HAIN: *English grapheme-to-phoneme conversion and evaluation*. Speech and Language Technology, 11:159–166, 2008.

[8] REICHEL, U. and F. SCHIEL: *Using Morphology and Phoneme History to improve Grapheme-to-Phoneme Conversion*. In *Proc. Eurospeech*, pp. 1937–1940, Lisboa, 2005.

[9] SURENDRAN, D. and P. NIYOGI: *Quantifying the functional load of phonemic opposition, distinctive features, and suprasegmentals*. In *Competing Models of Linguistic Change: Evolution and beyond. In commemoration of Eugene Coseriu (1921–2002)*, pp. 43–53. Benjamins, Amsterdam, Philadelphia, 2006.

[10] VAN DEN HEUVEL, H., J. MARTENS and N. KONINGS: *G2P conversion of names. What can we do (better)?*. In *Proc. Interspeech*, pp. 1773–1776, Antwerp, 2007.

[11] WAGNER, R. and M. FISCHER: *The string to string correction problem*. Journal of the Association for Computing Machinery, 21(1), 1974.