Similarity metrics for aligning children's articulation data

Harold L. SOMERS
Centre for Computational Linguistics
UMIST, PO Box 88,
Manchester M60 1QD, England
harold@ccl.umist.ac.uk

1. Background

This paper concerns the implementation and testing of similarity metrics for the alignment of phonetic segments in transcriptions of children's (mis)articulations with the adult model. This has an obvious application in the development of software to assist speech and language clinicians to assess clients and to plan therapy. This paper will give some of the background to this general problem, but will focus on the computational and linguistic aspect of the alignment problem.

1.1. Articulation testing

It is well known that a child's acquisition of phonology is gradual, and can be charted according to the appearance of phonetic distinctions (e.g. stops vs. fricatives), the disappearance of childish mispronunciations, especially due to assimilation ([gɒg] for *dog*), and the ability to articulate particular phonetic configurations (e.g. consonant clusters). Whether screening whole populations of children, or assessing individual referrals, the articulation test is an important tool for the speech clinician.

A child's articulatory development is usually described with reference to an adult model, and in terms of deviations from it: a number of phonological "processes" can be identified, and their significance with respect to the chronological age of the child assessed. Often processes interact, e.g. when *spoon* is pronounced [mun] we have consonant-cluster reduction and assimilation.

The problem for this paper is to align the segments in the transcription of the child's articulation with the target model pronunciation. The task is complicated by the need to identify cases of "metathesis", where the corresponding sounds have been reordered (e.g. *remember* → [mɪrɛmbə]) and "merges", a special case of consonant-cluster reduction where the resulting segment has some of the features of both elements in the original cluster (e.g. *sleep* → [ɬip]).

It would be appropriate here to review the software currently available to speech clinicians, but lack of space prevents us from doing so (see Somers, forthcoming). Suffice it to say that software *does* exist, but is mainly for grammatical and lexical analysis. Of the tiny number of programs which specifically address the problem of articulation testing, none, as far as one can tell, involve *automatic* alignment of the data.

1.2. Segment alignment

In a recent paper, Covington (1996) described an algorithm for aligning historical cognates. The present author was struck by the possibility of using this technique for the child-language application, a task for which a somewhat similar algorithm had been developed some years ago (Somers 1978, 1979). In both algorithms, the phonetic segments are interpreted as bundles of phonetic features, and the algorithms include a simple *similarity metric* for comparing the segments pairwise. The algorithms differ somewhat in the way the search space is reduced, but the results are quite comparable (Somers, forthcoming).

Coincidentally, a recent article by Connolly (1997) has suggested a number of ways of quantifying the similarity or difference between two individual phones, on the basis of perceptual and articulatory differences. Connolly's metric is also feature-based, but differs from the others mentioned in its complexity. In particular, the features can be differentially weighted for salience, and, additionally, not all the features are simple Booleans. In the second part of his article, Connolly introduces a distance measure for comparing *sequences* of phones, based on the Levenshtein distance well-known in the spell-checking, speech-processing and corpus-alignment literatures (*inter alia*). Again, this metric can be weighted, to allow substitutions to be valued differentially (on the basis of the individual phone distance measure as described in the first part), and to deal with merges and metathesis.

Although his methods are clearly computational in nature, Connolly reports (personal communication) that he has not yet implemented them. In this paper, we describe a simple implementation and adaptation of Connolly's metrics, and a brief critical evaluation of their performance on some child language data (both real and artificial).

2. The alignment algorithms

We have implemented three versions of an alignment algorithm, utilising different segment similarity measures, but the same sequence measure.

2.1. Coding the input

Before we consider the algorithms themselves, however, it is appropriate to mention briefly the issue of transcription. On the one hand, children's articulations can include a much wider variety of phones than those which are found in the target system; in addition, certain secondary phonetic features may be particularly important in the description of the child's articulation (e.g. spreading, laryngealization). So the transcriptions need to be "narrow". On the other hand, speech clinicians nevertheless tend to use a "contrastive" transcription, essentially phonemic except where the child's articulation differs from the target: so normal allophonic variation will not necessarily be reflected in the transcription. Any program that is to be used for the analysis of articulation data will need an appropriate coding scheme which allows a narrow transcription in a fairly transparent notation. Some software offers phonetic transcription schemes based on the ASCII character set (e.g. Perry 1995). Alternatively, it seems quite feasible to allow the transcriptions to be input using a standard word-processor and a phonetic font, and to interpret the symbols accordingly. For a commercial implementation it would be better to follow the standard proposed by the IPA (Esling & Gaylord 1993), which has been approved by the ISO, and included in the Unicode definitions.

2.2. Internal representation

Representing the phonetic segments as bundles of features is an obvious technique, and one which is widely adopted. In the algorithm reported in Somers (1979) — henceforth CAT — phones are represented as bundles of binary articulatory features. Some primary features also serve as secondary features where appropriate (e.g. dark 'l' is marked as VEL(ar)), but there are also explicit secondary features, e.g. ASP(iration).

Connolly (1997) suggests two alternative feature representations. The first is based on *perceptual* features, which, he claims, are more significant than articulatory features "from the point of view of communicative dysfunction" (p.276). On the other hand, he admits that using perceptual features can be problematic, unless "we are prepared to accept a relatively unrefined quantification method" (p.277). Connolly rejects a number of perceptual feature schemes for consonants in favour of one proposed by Line (1987), which identifies two perceptual features or axes, "friction strength" (FS) and "pitch" (P), and divides the consonant phones into six groups, differentiated by their score on each of these axes, as shown in Figure 1.

Henceforth we will refer to this scheme as "FS/P". In fact, there are a number of drawbacks and shortcomings in Connolly's scheme for our purposes, notably the absence of many non-English phones (all non-pulmonics, uvulars, retroflexes, trills and taps), and there is no indication how to handle secondary features typically needed to transcribe children's articulations accurately. We have tried to rectify the first shortcoming in our implementation, but it is not obvious how to deal with the second.

Connolly's alternative feature representation is based on *articulatory* features, adapted from Ladefoged's (1971) system, though unlike the features used in the CAT scheme, some of the features are not binary. Figure 2 shows the feature scheme for consonants, which we have adapted slightly, in detail. We will refer to this

Figure 1. Perceptual feature-based representation (FS/P) of consonants from Connolly (1997:279f)

Group	Friction-strength	Pitch	Members
1	0.0	0.0	bilabial plosives; labial and alveolar nasals
2	0.0	0.4	glottal obstruents; central and lateral approximants; palatal and velar nasals
3	0.4	0.3	alveolar plosives; labial and dental fricatives; voiceless nasals
4	0.5	0.8	velar and palatal obstruents
5	0.8	0.9	palato-alveolar and lateral fricatives
6	1.0	1.0	alveolar fricatives and affricates

Figure 2. Articulatory feature scheme (Lad) for consonants, adapted from Connolly (1997:282f).

(a) non-binary features with explanations of the values:

glottalic: 1 (ejective), 0.5 (pulmonic), 0 (implosive)
voice: 1 (glottal stop), 0.8 (laryngealized), 0.6 (voiced), 0.2 (murmur), 0 (voiceless)
place (i.e. passive articulator): 1 (labial), 0.9 (dental), 0.85 (alveolar), 0.8 (post-alveolar), 0.75 (pre-palatal), 0.7 (palatal), 0.6 (velar), 0.5 (uvular), 0.3 (pharyngeal), 0 (glottal)
constrictor: 1 (labial), 0.9 (dental), 0.85 (apical), 0.75 (laminal), 0.6 (dorsal), 0.3 (radical), 0 (glottal)
stop: 1 (stop), 0.95 (affricate), 0.9 (fricative), 0 (approximant)
length: 1 (long), 0.5 (half-long)

(b) binary features:

velaric (for clicks), aspirated, nasal, lateral, trill, tap, retroflex, rounded, syllabic, unreleased, grooved

scheme as "Lad". Again, some features or feature values needed to be added, notably a value of "stop" for affricates.

Let us now consider the similarity metrics based on these three schemes.

2.3. Similarity metrics for individual phones

The similarity (or distance) metric is the key to the alignment algorithm. In the case of CAT, the distance measure is quite simply a count of the binary features for which the polarity differs. So for example, when comparing the articulation [d] with a target of [st], the [s] and [d] differ in terms of three features (VOICE, STOP and FRIC) while [t] and [d] differ in only one (VOICE): so [d] is more similar to [t] than to [s].

In FS/P, the two features are weighted to reflect the greater importance of FS over P, the former being valued double the latter. To calculate the similarity of two phones we add the difference in their FS scores to half the difference in their P scores. If the two phones are in the same group, the score is set at 0.05 (unless they are identical, in which case it is 0). Thus, to take our [st]→[d] example again, since [s] is in group 6, and [t] and [d] both in group 3, [t]~[d] scores 0.05, [s]~[d] 0.95.

The similarity metric based on the Lad scheme is simpler, in that all the features are equally weighted. The Lad score is the simply sum of the score differences for all the features.

For our example of [st]→[d], the [t]~[d] difference is only in one feature, "voice", with values 0 and 0.6 respectively, while the [s]~[d] difference has the 0.6 voice difference plus a difference of 0.1 in the "stop" feature ([d] scores 1, [s] scores 0.9).

All three metrics agree that [d] is more similar to [t] than to [s], as we might hope and expect. As we will see below, the different feature schemes do not always give the same result however.

2.4. Sequence comparison

Connolly's proposed algorithm for aligning sequences of phones is based on the Levenshtein distance. He calls it a "weighted" Levenshtein distance, because the algorithm would have to take into account the similarity scores between individual segments when deciding in cases of combined substitution and deletion (e.g. our [st] → [d] example) which segment to mark as inserted or deleted. Connolly suggests (p.291) that substitutions should always be preferred over insertions and deletions, and this assumption was also built into the algorithm we originally developed in Somers (1979). However, this does not always give the correct solution: for example, if the sequence [skr] (e.g. in *scrape*) was realised as [ʃsk], we would prefer the alignment in (1a) with one insertion and one deletion, to that in (1b) with only substitutions.

```
(1) a. – s k r     b. s k r
       ʃ s k –        ʃ s k
```

The algorithm would also have to be adjusted to allow for metathesis, though Connolly suggests that merges do not present a special problem because they can always be treated as a substitution plus an omission (p.292) — again we disagree with this approach and will illustrate the problem below.

For these reasons we have not used a Levenshtein distance algorithm for our new implementation of the alignment task. As described in Somers (forthcoming), the original alignment algorithm in CAT relied on a single predetermined anchor point, and then exhaustively compared all possible alignments either side of the anchor, though only when the number of segments differed.

We now prefer a more general recursive algorithm in which we identify in the two strings a suitable anchor, then split the strings around the two anchor points, and repeat the process with each half string until one (or both) is (are) reduced to the empty string. The algorithm is given in Figure 3. Step 2 is the key to the algorithm, and is primed to look first for identical phones, else vowels, else the phones are compared pairwise exhaustively. If there is a choice of "best match", we prefer values of *i* and *j* that are similar, and near the middle of the string. Although the algorithm is looking for the best match, it is also looking for possible merges, which will be identified when there is no single best match.

2.5. Identifying metathesis

It is difficult to incorporate a test for metathesis directly into the above algorithm, and it is better to make a second pass looking for this phenomenon explicitly. For our purposes it is reasonable to focus on consonants. Metathesis can occur either with contiguous phones, e.g. [dɛsk] → [dɛks], or with phones either side of a vowel, e.g. [ɛlɪfənt] → [ɛfɪlənt]. In addition, one or both of the phones may have undergone some other phonological processes, e.g. [ɛlɪfənt] → [ɛpɪlənt], where the [f] and [l] have been exchanged, but the [f] realised as a [p].

Figure 3. The alignment algorithm.

```
Let X and Y be the strings to be aligned, of length $m$ and $n$, where each X[$i$], Y[$j$], $1 \leq i \leq m$, $1 \leq j \leq n$, is a bundle of features.
1. If X=[] and Y=[], then stop; else if X=[]
   (Y=[]) then mark all segments in Y (X) as
   "inserted" ("omitted" ) and stop; else
   continue.
2. Find the best matching X[$i$] and Y[$j$], and
   mark these as "aligned".
3. Take the substring X[1]..X[$i$–1] and the
   substring Y[1]..Y[$j$–1] and repeat from step
   1; and similarly with the substrings
   X[$i$+1]..X[$m$], and Y[$j$+1]..Y[$n$].
```

The algorithm described above will analyse metatheses in one of two ways, depending on various other factors. One analysis will simply align the phones with each other. To recognise this as a case of metathesis, we need to see if the crossing alignment gives a better score. The other analysis will align one or other of the identical phones, and mark the others as omitted/inserted. The second pass looks out for both these situations.

3. Evaluation

In this section we consider how the algorithm deals with some data, both real and simulated. We want (a) to see if the algorithm as described gets alignments that correspond to the alignment favoured by a human; and (b) to compare the different feature systems that have been proposed.

For many of the examples we have used, there is no problem, and nothing to choose between the systems. These are cases of simple omission (e.g. *spoon*→[pun]), insertion (*Everton* → [ɛvətʌnt]), substitution (*feather* → [bɛɣə]), and

→ [ɛvətʌnt]), substitution (*feather* → [bɛɣɔ]), and various comḅinations of those processes (*Christmas*→[gɪxməx], *aeroplane*→[wɛjəbein]). Cases of inserted vowels (e.g. *spoon*→[supun]) were analysed correctly when the inserted vowel was different from the main vowel. So for example *chimney* → [tʃɪmɪnɪ] caused difficulty, with the alignment (2a) preferred over (2b).

(2) a. tʃ ɪ m n ɪ – –   b. tʃ ɪ m – n ɪ
      tʃ ɪ m – ɪ n ɪ   tʃ ɪ m ɪ n ɪ

Differences between the feature systems show up when the alignment combines substitutions and omissions, and the "best match" comes into play. Vocalisation of syllabics (e.g. *bottle* [bɒtɫ̩] → [bɒʔuw]) caused problems, with the syllabic [ɫ̩] aligning with [u] in the CAT system, [ʔ] in FS/P, and [w] in Lad.

In other cases where the systems gave different results, the FS/P system most often gave inappropriate alignments. For example, *monkey* [mʌŋki] → [mʌnʔi] was correctly aligned as in (3a) by the other two systems, but as (3b) with FS/P.

(3) a. m ʌ ŋ k i   b. m ʌ – ŋ k i
      m ʌ n ʔ i   m ʌ n ʔ – i

For *teeth* [tiθ]→[ʔisx], FS/P aligned the [x] with the [θ] while the other systems got the more likely [θ]→[s] alignment. Similarly, the Lad and CAT systems labelled the [ɹ] as omitted in *bridge* [bɹɪdʒ]→[gɪx], while FS/P aligned it with [g].

When identifying merges on the other hand, only CAT had any success, in *sleep* [sl̥ip]→[ɬip] (but not when the [l] is not marked as voiceless). In analysing [fl]→[b], CAT suggests a merge, FS/P marks the [f] as omitted, Lad the [l]. In principle, the FS/P system offers most scope for identifying merges, as it only recognises six different classes of consonant phone, while the Lad system is too fine-grained: indeed, we were unable to find (or simulate) any plausible case which Lad would analyse as a merge.

Against that it should also be noted that such analyses cannot be carried out totally in isolation. For example, compare the case where [ɬ] is only used when [sl] is expected to the one where [s] is generally realised as [ɬ]: we might want to analyse only the former case as a merge, the latter as a substitution plus omission. It should be remembered that the alignment task is only the first step of the analysis of the child's phonetic system.

4. Conclusion

Because of its poor performance with many alignments, we must reject the FS/P system. This is not a great surprise: a feature system based on perceptual differences seems intuitively questionable for an *articulation* analysis task. There does not seem much to choose between Lad and CAT, though the former gives a more subtle scoring system, which might be useful for screening children. On the other hand, it never identifies merges, even in highly plausible cases, so the system using simpler binary articulatory features may be the best solution.

Whichever system is used, it seems that an acceptable level of success can be achieved with the algorithm described here, and it could form the basis of software for the automatic analysis of children's articulation data.

5. References

Connolly, John H. (1997) Quantifying target–realization differences. *Clinical Linguistics & Phonetics* 11:267–298.

Covington, Michael A. (1996) An algorithm to align words for historical comparison. *Computational Linguistics* 22:481–496.

Esling, John H. & Harry Gaylord (1993) Computer codes for phonetic symbols. *Journal of the International Phonetic Association* 23:83–97.

Ladefoged, P. (1971) *Preliminaries to Linguistic Phonetics*. Chicago: University of Chicago Press.

Line, Pippa (1987) An Investigation of Auditory Distance. M.Phil. dissertation, De Montfort University, Leicester.

Perry, Cecyle K. (1995) Review of Phonological Deviation Analysis by Computer (PDAC). *Child Language Teaching and Therapy* 11:331–340.

Somers, H.L. (1978) Computerised Articulation Testing. M.A. thesis, Manchester University.

Somers, H.L. (1979) Using the computer to analyse articulation test data. *British Journal of Disorders of Communication* 14:231–240.

Somers, H.L. (forthcoming) Aligning phonetic segments for children's articulation assessment. To appear in *Computational Linguistics*.

Similarity metrics for aligning children's articulation data

An important step in the automatic analysis of child-language articulation data is to align the transcriptions of children's (mis)articulations with adult models. The problems underlying this task are discussed and a number of algorithms are presented and compared. These are based on various similarity or distance measures for individual phonetic segments, considering perceptual and articulatory features, which may be weighted to reflect salience, and on sequence comparison.

子どもの発音のデータを比較する類似性測定法

子どもの発音データの自動分析における重点段階は、子どもの発音を大人のモデル音と音声比較することで、測定することにある。自動分析作業に関わる問題点を論ずるにあたり、いくつかのアルゴリズムが比較評価されている。このデータは各々の音声節の類似性や相違点に根差して測定されたものである。聴覚特徴と発音特徴のアルゴリズムが比較検討されている。単純なる発語置き換え、例えば、「りんご」を「ディンゴ」という、などは分析しやすいので、アルゴリズムの選択は必要ない。しかしもっと複雑な発音の場合は、アルゴリズムの測定結果が多岐にわたる。単純なる発音特徴のアルゴリズムに一番いい結果が見られた。

Acknowledgements

Thanks to Joe Somers for providing some of the example data; and to Marie-Jo Proulx and Ayako Matsuo who helped with the abstracts.

Une comparaison de quelques mesures de ressemblance pour l'analyse comparative des transcriptions d'articulation infantile

En ce qui concerne l'analyse des transcriptions d'articulation infantile, il est très important d'identifier les correspondences entre les articulations de l'enfant, parfois fausses, et celles de l'adulte perçues en tant que modèle. Nous décrivons l'automatisation de cette tâche, et présentons quelques algorithmes dont nous faisons une comparaison évaluative. Les algorithmes se basent sur certaines mesures de ressemblance (ou distance) phonétique entre les segments individuels qui considèrent les traits perceptuels et articulatoires, ceux qui peuvent porter des poids selon leur saillance. Il s'agit aussi d'une comparaison de séquences.

Les erreurs d'articulation sont parfois de simples substitutions d'un son par un autre, ou des insertions ou omissions, qui sont faciles à analyser. Les problèmes découlent surtout des "métathèses" (par ex. *éléphant* s'exprime [efelã]), surtout où il y a aussi une substitution (par ex. [epelã] pour *éléphant*), et des "fusions" (par ex. *crayon* [kʀɛjɔ̃] → [xɛjɔ̃]) où le [x] rassemble également au [k] et au [ʀ].

Les trois mesures de ressemblance utilisent les traits phonétiques: un système de simples traits articulatoires binaires (TAB) élaboré par le présent auteur; un système de traits perceptuels ("force de friction" et "ton" FF/T) élaboré par Connolly (1997); et un système de traits articulatoires non-binaires basé sur Ladefoged (1971). Pour beaucoup d'exemples, les trois systèmes ont trouvé la même solution. Là où ils diffèrent, le système FF/T est moins performant. Entre les deux autres, le système le plus simple (TAB) semble aussi être le plus robuste. Pour la comparaison des séquences, un seul algorithme est présenté. Il fonctionne très bien, sauf quand il s'agit d'une voyelle identique insérée (par ex. [kʀɛjɔ̃] → [kɛʀɛjɔ̃]).

Parmi les logiciels commercialisés destinés aux orthophonistes actuellement disponibles, aucun ne comprend d'analyse automatique des articulations, celle-ci étant considérée "trop difficile". Le présent travail suggère qu'un tel logiciel est au contraire tout à fait concevable.