THE EFFECT OF LINGUISTIC EXPECTANCY ON PHONETIC TRANSCRIPTION: DEVELOPING AN ADEQUATE ALIGNMENT ALGORITHM

C. Cucchiarini* & R. van Bezooijen**

* Department of Language and Speech, Nijmegen, The Netherlands
** Institute of Phonetic Sciences, Amsterdam, The Netherlands

ABSTRACT

This paper describes an alignment algorithm developed for transcription comparison. Theoretical and practical problems connected with the use of such a program are considered (1).

1. INTRODUCTION

A segmental transcription is the auditory analysis of an utterance into discrete units of sound represented by phonetic symbols. Such an analysis may be undertaken either to give a very detailed description of an utterance (allophonic transcription) or to indicate the distinctive categories of a language (phonemic transcription). Implicit in this distinction is the notion that the transcription is made by a transcriber who is familiar with the language to be transcribed. A different type of transcription may be obtained when a transcriber is required to transcribe an unknown language. The result is a so-called impressionistic transcription. The term impressionistic here refers to the fact that the transcriber has no recourse to the phonological system of the language being transcribed.

All three types of transcription, i.e. allophonic, phonemic, and impressionistic, have long been used in many fields of linguistic research as a means of recording speech material. However, the validity of these procedures has hardly ever been questioned. This is surprising, especially if we consider that analyses of this type are subject to the influence of a great number of variables relating both to the transcriber (experience, degree of familiarity with the language being transcribed, concentration, auditory acuity etc.) and to the type of speech under investigation (speech style, length of the utterance, rate of speech etc.).

In the light of these considerations we thought it would be useful to determine to what extent transcription performance can vary as a function of some of the factors mentioned above. Three variables were selected for investigation: 1. the transcriber's degree of familiarity with the language transcribed, 2. the presence of linguistic context, and 3. speech style. What these three variables have in common is that they are all related to linguistic expectancy, albeit to different degrees.

In the following section we will describe the method used, paying particular attention to the alignment program developed for transcription comparison and to the problems associated with the use of such a program. In section 3 preliminary results of its application will be presented.

2. METHOD

2.1. Transcription alignment

In order to determine the effect of the above-mentioned factors on transcription performance we need to be able to measure the difference between two transcriptions of the same utterance. Since phonetic transcriptions are linear sequences of symbols, the overall difference between two transcriptions of the same utterance is here defined as the sum of the differences between corresponding elements, i.e. symbols describing the same articulatory event. This implies that before two strings of symbols can be compared they have to be aligned, i.e. each symbol in one string has to be matched with the corresponding symbol in the other string.

Considering the enormous amount of material in our investigation (8640 transcriptions to be compared thousands of times) it was unthinkable to align transcriptions by hand. A program was therefore developed which makes it possible to automatically align different transcriptions of the same utterance. The algorithm employed in our alignment program very much resembles the one developed by Picone et al. [2]. This is an adapted version of the standard dynamic programming algorithm, which aligns two strings of symbols minimizing the cumulative distance between them [1]. String alignment is performed on the basis of distance measures between symbols. If the two transcriptions do not contain the same number of phonetic symbols, null symbols are inserted. On the basis of the distance values, the alignment program determines which symbols are missing in one of the two transcriptions (or have been inserted in the other, depending on the point of view). Owing to space limitations, we cannot go into the difficulties involved in deriving the distance values for transcription evaluation. These difficulties concern not only the choice of the numerical values, but first and foremost the choice of the domain in which speech sounds are to be compared, i.e. perception, acoustics or articulation. Suffice it to say that for want of a better solution we eventually decided to use two matrices, one for vowels and one for consonants, in which sounds are defined by feature values [3]. The features adopted are essentially articulatory. This choice was primarily motivated by the fact that phonetic symbols are defined in terms of articulatory characteristics.

The major differences between our program and that of Picone et al. concern the input matrix:

1. Picone et al. use phoneme distance matrices while our program employs matrices containing feature values. The distances between speech sounds are computed as the program needs them. Although this makes the system slower, it has the important advantage of making it possible to include diacritical marks. Their effect on the different phonetic symbols is computed before determining the distance between two basic symbols.
2. The matrices adopted by Picone et al. contain perceptually based distances, whereas our features are essentially articulatory.
3. Apart from a few exceptions, both programs disallow vowel-to-consonant matches. In Picone et al. this is achieved by adding an extra matrix in which distances between vowels and consonants are greater than distances to the null symbol. A restricted number of matches between vowels and consonants is allowed by defining their distance to be lower than the distance to the null symbol. In our program vowel-to-consonant matches are prevented by rule. Possible exceptions are to be included in a separate list with their respective costs.

At best, an alignment program will perform as well as a human expert [1]. Of course human performance does not mean a hundred per cent correctness, as there can be string pairs which are simply difficult to align, even for an experienced phonetician. This may be the case when two transcriptions are very different, both quantitatively (number of symbols contained) and qualitatively (nature of the phonetic symbols).

When phoneticians align transcriptions by hand they use their knowledge of speech production and perception to arrive at what they think is the best alignment. Alternatively, when an automatic system is used this knowledge has to be externalized in the form of rules, constraints or costs, which tell the alignment program what to do. It is evident that even the best combination of rules and distance values cannot guarantee the performance level of a human expert, as the latter has access to much more information, can use his intuitions and can be more flexible. In other words, we have to settle for something which can only approach human performance. This means that in any alignment program human corrections will eventually be required.

When an alignment program produces unsatisfactory output there are two possible solutions: 1. one can alter the output or 2. one can change the structure of the program (rules and distance values). Although the first solution would be the easiest, it is extremely ad hoc. Morever, it may be argued that if the program provides an undesirable solution it does so on the basis of the knowledge built in it. So, instead of manipulating the outcome

one should change the information which led to it. This would imply using the alignment program diagnostically to check whether the distance values are well chosen. For example, if the two following transcriptions are aligned as in 1 while we want the alignment to be as in 2,

```
1   d E n t        2  d E n t
    d E 0 m           d E m 0
```

then it is clear that the distance value between /t/ and /m/ is too small in relation to that between /n/ and /m/. Also changing the distance values has its drawbacks. Theoretically, it is not correct since distance values are based on feature counting and therefore have their own motivation. From a more practical point of view, there should be no objection to using the outcome of the alignment program in order to improve the distance matrices, as we know them to be far from ideal.

With null symbols things are different. In this case, feature counting cannot be applied simply because null symbols have no features. As a consequence, the distance value between a phonetic symbol and a null symbol can only be motivated by the efficiency of the alignment program: as long as the alignment is correct the null symbol values are also correct.

In the following section we will present some results of the application of our alignment program.

3. ADEQUACY OF THE ALIGNMENT PROGRAM: PRELIMINARY RESULTS

So far, the alignment program described above has been tested on 1680 transcription pairs. These were transcriptions made by fourteen Language and Speech Pathology students at the University of Nijmegen, in two experimental rounds. The material transcribed in the first round consisted of 120 speech fragments containing sequences of sounds across word boundaries, extracted from their original contexts so that they sounded like nonsense syllables. The fragments differed with respect to language variety (Dutch, a Dutch dialect, and an unknown language, viz. Czech) and speech style (reading vs. spontaneous speech). The material transcribed in the second round consisted of the same fragments, this time presented in their original contexts (usually two or three words). The transcriptions were made in accordance with the pre-1989 version of the IPA.

As mentioned above, null symbols constitute a problem because one simply does not know what value they should be assigned. Initially, we gave null symbols maximum values, computed on the basis of the distances between phonetic symbols. So, for vowel deletion we obtained a value of 10 and for consonant deletion a value of 15. This choice turned out to be not very felicitous for two reasons, one theoretical, the other practical. First, it is not clear why deleting a consonant should have a higher value than deleting a vowel. Second, when used as input to the alignment program these values produced a few instances of distorted alignment, in that matching null symbols with vowels led to a smaller cumulative distance than matching them with consonants. In a second trial we adopted the value 15 for both vowels and consonants. As the alignment program aims at minimizing the cumulative distance between two strings, giving null symbols such a high value may result in alignments with an insufficient number of null symbols. Conversely, lower values may lead to alignments with too many null symbols.

In order to get a general idea of how our program works we checked all alignments obtained to determine whether they were correct. Cases of incorrect alignment were classified as follows:

1. incorrect alignment due to an insufficient number of null symbols
2. incorrect alignment due to the insertion of too many null symbols.
3. incorrect alignment due to incorrect distance values between segments
4. difficulty in finding the right correspondence between the two strings

Out of a total number of 1680 string pairs, 87 (5.17%) turned out to be incorrectly aligned. The distribution observed was the following:

Table 1. Incorrect alignments

error type	1	2	3	4
cases	7	71	3	6

As is clear from this table the number of incorrect alignments of the second type is disproportionately high. This has two main causes. The first, which accounts for 52 cases, is the impossibility of matches between vowels and consonants. We expected this to be a problem and had already planned to use a list of exceptions (see section 2.1.) First, however, we wanted to get an idea of the incidence of these cases. Now the question is whether the exceptions should be included in the program, which could have undesirable results for other string pairs, or whether they should be applied afterwards.

The second cause, which accounts for 19 cases, is the incorrect matching of diphthongs with long vowels. In its present form, the program aligns the long vowel with the first part of the diphthong and then matches the second part with a null symbol. Since this appear counterintuitive it will have to be changed by making it possible to match the whole of the diphthong with the long vowel.

Apart from these cases, for which a solution has already been suggested, the number of incorrect alignments is small (0.95%). This would seem to indicate that, with the improvements proposed above, the program should work satisfactorily.

At this point another crucial question arises: are the distance values used for transcription alignment to be used also as an indication of error gravity? This question particularly concerns the values attributed to null symbols. For instance, in our case the extremely high cost associated with null symbols led to satisfactory alignments, but it also had the effect of strongly influencing the average distance between transcriptions computed by the alignment program (for vowels and consonants separately). In fact, the transcription pairs with the highest dissimilarity scores were those in which null symbols had been inserted. In order to gain more insight into the effect of the null symbol value on transcription alignment we let the program align the same transcriptions again, but this time with an average value for null symbols, viz. 7. This led to exactly the same distribution as that presented in table 1. Obviously, the value 7 is to be preferred to 15 because it has less impact on the distance measure and still produces a high proportion of correct alignments. Even this lower value, however, has the effect of penalizing null symbol insertion. Of course this need not be wrong. If one thinks that omitting segments or inserting them is a serious mistake then it is right to associate a high cost with null symbol insertion. Perhaps one would like to introduce gradations in the cost of deletions, so that omitting certain segments is considered more serious than omitting others. In general, one cannot a priori exclude the possibility that under certain circumstances it may be appropriate to adopt different values for transcription alignment and transcription evaluation. Each case will have to be considered separately and the outcome will depend on the purpose of the transcription.

4. REFERENCES

[1] KRUSKAL, J.B. & D. SANKOFF (eds.) (1983), *"Time warps, string edits, and macromolecules: the theory and practice of sequence comparison"*, Reading (Mass.): Addison-Wesley Publishing Company.

[2] PICONE J., K.M. GOUDIE MARSHALL, G.R. DODDINGTON, & W. FISHER (1986), "Automatic text alignment for speech system evaluation", *IEEE Transactions on acoustics, speech, and signal processing, Vol. ASSP-34, No. 4*, 780-784.

[3] VIEREGGE, W.H. & C. CUCCHIARINI (1988), "Evaluating the transcription process", in: Ainsworth, W.A. & J.N. Holmes (eds.) *Proceedings Speech '88*.

(1) This research was supported by the Foundation for Linguistic Research, which is funded by the Netherlands organization for research, NWO.