ARIPUC 16, 1982, p. 127-136

THE CONSTRUCTION OF A GRAPHEME-TO-PHONE ALGORITHM FOR DANISH

PETER MOLBÆK HANSEN

The paper presents the main strategy and some preliminary results of the current work with the construction of a grapheme-to-phone algorithm forming part of a text-to-speech algorithm for Danish. The paper also discusses the main difficulties connected with this work, and a brief report of the present status of the project is given.

I. INTRODUCTION

This paper reports on the ongoing work with the construction of a grapheme-to-phone or rather text-to-phonetic representation algorithm forming part of the text-to-speech project for Danish on which more general information is given by Peter Holtse elsewhere in this report.

The paper is divided into three main sections. In section II the main features of the algorithm are presented. In section III some details of the orthographic component of the algorithm are described, and examples of the interplay between rules and exceptions are given. In section IV a few remarks on current and future work are given.

II. GENERAL DESCRIPTION OF THE ALGORITHM

In this section the main features and components of the grapheme-to-phone algorithm (henceforth GTPA) will be described. General, language-independent problems connected with the construction of text-to-speech algorithms will not be mentioned directly. A concise overview with useful references is found in Sherwood (1981).

A. REGULARITIES AND EXCEPTIONS

The algorithm should be mainly rule-oriented, i.e. both orthographic and phonological phenomena which can in any reasonable sense be described as regularities should be stated as formal transformation rules not referring to the identity of single words.

The main problems that arise when one attempts to apply this requirement to Danish have to do with the fact that Danish orthography has serious drawbacks from the point of view of its relation to pronunciation. As is well known, it is not the case that Danish orthography is consistent in the sense that the pronunciation of any given word can be derived from its spelling by a limited set of exceptionless rules governing the correspondence between letters (letter combinations) and sounds (sound combinations). If that were the case, it would be easy to make a fail safe (and efficient, cf. section III) GTPA at word level. Danish orthography does not even meet the much weaker requirement that the pronunciation of any given word be inferable from its spelling in a unique, if not necessarily simple way, cf. the existence of heterophonic homographs in which - paradoxically enough for an alphabetic writing system - Danish orthography abounds (cf. e.g., the words [hu:'l] 'hollow' and [hɔl] 'hole' which are both spelled *hul*, see further section III). If the latter requirement were met, a fail safe GTPA could still in principle be constructed, although it would have to cope with difficulties of the kind to be discussed in section III.

The main strategy must take as its point of departure that Danish orthography is nevertheless rich in partial regularities as regards the correspondence between spelling and pronunciation. The existence of the above mentioned drawbacks only means that a GTPA for Danish can neither be 100% fail safe nor consist exclusively of exceptionless rules.

The strategy underlying the construction of the GTPA is based upon a specific interpretation of two general (and, of course, well known) concepts: regularities and exceptions. These two concepts are in turn applied to two different areas: the area of feature assignment to letters, and the area of letter-to-segment-transformation.

B. THE TWO COMPONENTS

The general principles underlying the GTPA - the construction of which is far from completed - are shown in outline in figure 1. There are two components, a component taking care of exceptional spellings (henceforth XCO) and a rule component (henceforth RCO). The task of the XCO is to take care of exceptional spellings and change them into spellings which are consistent with the rules of the RCO in the sense that these will apply in an unrestricted way to the output of the XCO and yield a correct output. This also implies that the XCO supplies information

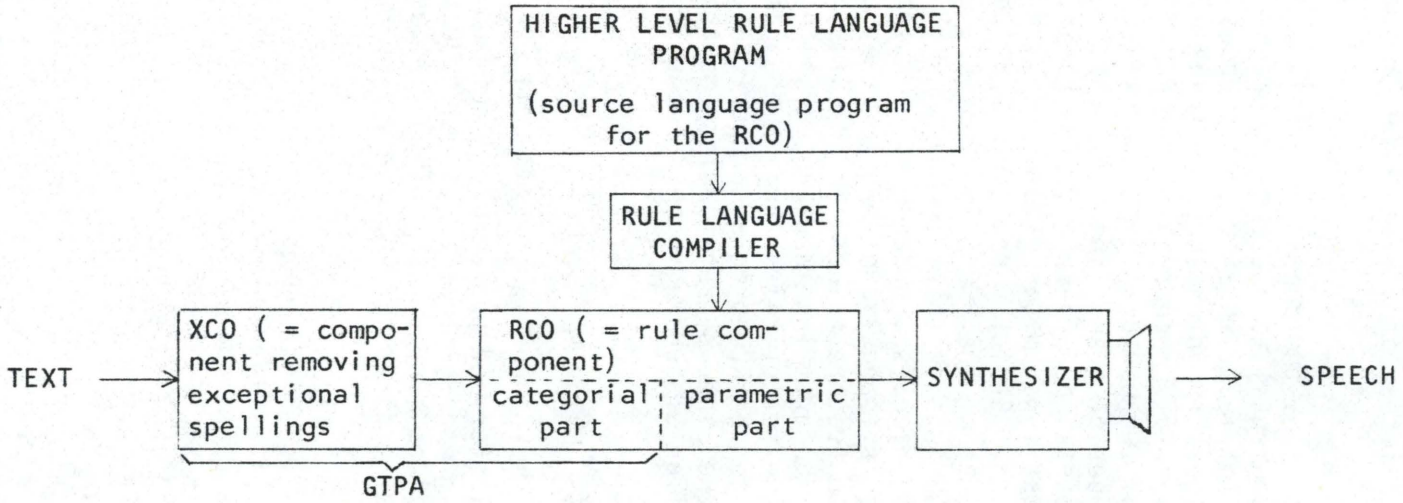

Figure 1

Schematic representation of the components of the text-to-speech-system.

which is necessary in order to predict the pronunciation of the sentence even if such information is not directly extractable from the orthographic form of the sentence (in particular information on syntactic and/or prosodic properties of the sentence). The output of the XCO will be referred to as "consistent notation".

The XCO takes an orthographic input, one sentence at a time, transforms it to consistent notation, assigns a bundle of binary feature values to each letter of the sentence, and delivers the output to the RCO.

The RCO transforms its input (i.e. the consistent notation with feature values assigned to segments) to a set of numeric values organized in a way which is (at least ideally) isomorphous with the relevant acoustic parameters to which they are translated in the speech synthesizer proper. Conceptually, the RCO may be described as consisting of two parts: 1) a categorial part dealing with an integral number of quasi-linguistic entities, viz. prosodic elements, segments, and features; 2) a parametric part dealing with quasi-acoustic parameter values to which the categorial elements are mapped. Strictly speaking, only the first of these conceptual parts of the RCO is part of the GTPA. Technically, however, the RCO is planned to be one monolithic algorithm, viz. a set of exceptionless, ordered rules which all have equal formal status irrespective of whether they apply to parameter values or to feature values. Such rules are to be formulated in a higher programming language, a "rule language"; in other words, the set of rules forms a program which in turn is translated to the RCO-program proper by means of a special compiler (cf. the paper by Peter Holtse on the construction of a rule compiler elsewhere in this report).

C. THE CONCEPT OF "INITIAL VALUE"

It is an important property of the RCO that there will be no formal distinction between letter and phonological segment ("phonological segment" being used here as a common denominator for any bundle of feature values irrespective of whether it is near the upper end (the input end) or the lower end (the output end) of the algorithm).

It is nevertheless clear that some of the early rules must take care of what may be thought of as letter-to-systematic phoneme rules or the like, namely in cases where a letter corresponds to more than one segment. In such cases a strategy of "optional initial value" will be applied. This strategy may best be illustrated by an example: The letter *e* has three main phonological correspondences: /e:/ as in *del* 'part', /ɛ/ as in *let* 'easy', and /ə/ as in *gade* 'street'. If *e* is assigned the initial value of one of these segments, two rules must take care of assigning it the other values in the appropriate contexts. Thus from the point of view of number of rules, the three potential initial values will be equally costly. Since rules of this kind are not phonological rules in a traditional

sense, the decision of which value to choose for the letter *e* must be based on other considerations. Here considerations of program efficiency should be allowed to play a role. In phonological derivations of SPE type a rule is not generally evaluated from the point of view of its actual applicational burden, i.e. the actual number of times the structural description of a rule is met when that rule tries to apply to actual running text. Such considerations are of great importance when the algorithm is to be implemented by a computer program.

In the actual example there is absolutely no doubt that from the point of view of efficiency the initial value of *e* should be (the feature composition of) /ə/: /ə/ is by far the most common value of *e*, and the sum of successful applications of rules changing *e* to /e:/ and /ɛ/ in any normal Danish text would be far less than the sum of successful applications of one of these rules and a rule changing *e* to /ə/ which would be needed if either /e:/ or /ɛ/ were chosen as the initial value of *e*.

In most cases of a one-to-many relation between letter and phonological segment it is not conspicuous, however, which one of several (in most cases only two) possible initial values will yield the lowest actual applicational burden. It is a simple task, however, to have the program itself count the number of successful applications of any rule; thus it is possible to arrive at a non-arbitrary choice between alternative solutions by applying alternative versions of the algorithm to a representative (preferably long) text.

D. THE USE OF "FEATURES"

For each letter/segment a high number of binary features is available. This gives a high degree of freedom in assigning feature values to a letter/segment. It is obvious that no phonological system as such will use all these features; this means that one can define and use features for other than phonological purposes. One can, e.g., define a feature ± dummy to characterize certain letters which for some reason are best treated as a group. Of course one is free to use the same features in later stages of the RCO in some phonologically or phonetically interpretable way. In section III the "orthographic" use of a dummy feature will be exemplified.

III. ORTHOGRAPHIC PROBLEMS

In this section the main problems with the XCO and their possible solutions will be discussed. The main task of the XCO is to take care of exceptions. If a word or a portion of a word exhibits a spelling which, given a set of rules, will be incorrectly processed by the RCO, that word or portion of a word must be changed in some way or other.

A. ONE-ROOT WORDFORMS

It is clear that one of the main problems connected with the construction of an XCO is the fact that Danish - like other Germanic languages - is characterized not only by the existence of a great deal of more or less lexicalized compounds, but also by the relative freedom to form new compounds from lexical material. In this section I shall restrict myself to illustrate some of the problems which remain if we presuppose that the compound problem can be solved in some way or other. In other words, I shall refer only to problems connected with one-root-wordforms.

One-root-wordforms by and large have the following structure: optional prefix + root + optional suffix + optional inflexional ending which can be either one of the consonants *s* and *t* or a mono- or disyllabic structure with schwa as the vowel(s).

Many of the orthographic irregularities of such wordforms are cases where a letter combination at the beginning or at the end of a root is identical to the spelling of a prefix or a suffix: in a word like *afsløre* 'reveal' the initial *af-* is the common high frequency prefix /av/ in which the phoneme /v/ is spelled *f*. It is absolutely clear that the great majority of words beginning with *af-* are /av/-prefixed words. This calls for the following rule in the RCO:

#af → #av

Since, however, rules in the RCO are supposed to be exceptionless, the XCO must take care of a handful of words in which initial *af-* belongs to the root and is pronounced /af/: *afasi*, 'aphasia', *aften* 'evening', and a few others. If these words were handed over to the RCO without change, they would be incorrectly processed by the above mentioned rule. The XCO must make these words immune, as it were, to that rule. One way of doing this is to have their *f* changed to something else, e.g. a dummy segment *F*, whose feature composition could be identical to that of *f* except for one dummy-feature for which "normal" segments have minus, whereas dummy segments like *F* have plus.

In order to make such a change the XCO must identify the words to be changed, and this involves certain problems. A root like *afasi* may occur in inflected forms like *afasien*, *afasiens*, *afasier*, *afasiers*, *afasierne*, *afasiernes*, the various inflexional endings being obviously irrelevant to the phenomenon under consideration. Only as much of a wordform as will uniquely identify the root, should appear in the exception list, and in most cases this means that only one entry per root need to be listed. In the case of *afasi*, however, this is not entirely fail safe: it is entirely possible to form a word like *afasiatisere* 'to make less Asiatic in type', in which *af-* is the /av/-prefix. One way of solving this problem is to have two entries in a left-end exception list, i.e. a list of exceptions which are scanned by the XCO from the beginning of the words: *afasi#* and *afasie*, the latter entry taking care of

the inflected forms. Incidentally, such a solution is a good illustration of the need for analyzing compounds, since a compound like *afasiproblemer* would only be correctly processed if it appeared as *afasi#problemer* at this stage of the XCO.

As a parallel example concerning the final portion of roots, consider wordforms ending in *-erne*. The vast majority of wordforms ending in this letter combination are definite plurals of nouns. But in words like *cisterne* 'cistern', *lanterne* 'lantern' and a few others, *-erne* belongs to the root and is pronounced [ɛrnə], whereas in the "normal" case, i.e. in definite plurals, both vowels are schwa. Such words, or rather as much of the final portion of such words as will uniquely identify the root, may be listed in a right-end list of exceptions together with information specifying what the root should be changed to in order for it to be processed correctly by the RCO.

At the moment left-end and right-end exception lists appear as data structures organized as arrays of pairs of character strings, the first member of each pair being the orthographic identifier of a root, and the second member being the corresponding exchange-string. The pairs are sorted alphabetically according to their first member so as to allow for a fast binary search algorithm.

B. VOWEL LENGTH IN STRESSED SYLLABLES

It is not always clear what to call a rule and what to call an exception to a rule. In word final stressed syllables the vowel is in most cases short if followed by more than one consonant, cf. *vask* 'wash', *mælk* 'milk', *vand* 'water'; however, in some cases - most notably in perfect participles of verbs - it is long: *vist* 'shown', *målt* 'measured'. If it is followed by one consonant only, the vowel is either long or short, cf. *sal* 'hall', *lys* 'light', *ben* 'leg', *fatal* 'fatal' with long vowel, and *bal* 'ball', *kys* 'kiss', *ven* 'friend', *metal* 'metal' with short vowel. In the former case there is a clear statistic dominance of wordforms with short vowel, whereas in the latter case the number of wordforms with long vowels is approximately balanced by the number of wordforms with short vowel. Irrespective of whether short or long vowel before a single consonant is chosen as the normal case, many words must be listed as exceptions. One might, of course, base a decision upon a frequency investigation of the two types in running text, but it seems more insightful to exploit the fact that in the overwhelming majority of nonfinal stressed syllables the vowel is long if followed by one consonant and short if followed by more than one consonant. That is, we may state a general rule that lengthens vowels before a single consonant. This implies, of course, that a few polysyllables and many monosyllables with a short vowel followed by one consonant and many words with a long vowel followed by two or more consonants must be listed as exceptions and supplied with appropriate exchange-strings causing them to be treated correctly by the RCO. In this

case the exchange-strings can be used to take care of another problem. It is a general rule that stressed word final syllables with short vowel take stød if their vowel is followed by a sonorant consonant which in turn is followed by a consonant letter, irrespective of whether or not the last of these consonant letters actually corresponds to a pronounced consonant, cf. *skovl* [sgʌu'l] 'shovel' and *vand* [van'] 'water'. Some of the words which have to be listed as exceptions because they have a short stressed vowel before a single consonant letter have stød, others do not have stød. One way of combining information about stød and length in the exchange-strings assigned to such exceptions would be to distinguish them from each other as well as from words with a long vowel by (1) supplying the stødless exceptions with an *h* after the vowel letter (thus rendering them immune to stød assignment, since *h* is voiceless and thus cannot take stød, and at the same time preventing them from having their vowel lengthened because of the cluster), and (2) supplying the stød-syllables with an *h* after the postvocalic sonorant consonant letter if the vowel is to be short (thus making these syllables meet the structural description of the stød assignment rule, yet rendering them immune to the vowel lengthening rule). For instance, *hav* [hau] 'sea', *sov* [sʌu?] 'slept', *tal* [tal] 'number', and *ven* [vɛn] 'friend' might be listed as exceptions and supplied with the exchange-strings *hahv*, *sovh*, *tahl*, and *vehn*, respectively. A rule eventually deleting *h* in such positions would then be needed in the RCO.

It must be borne in mind that the task of the XCO is primarily of a nonlinguistic nature and that all sorts of shortcuts and hocuspocus devices should be allowed in this part of the algorithm. This is also essential from the point of view of making the XCO a fast and efficient tool: Since most of these exceptions are concerned with a word final syllable, they should probably be listed in the right-end exception list so as to minimize the computer time needed to compare - one character at a time - the text words with the items of the list.

At the moment an operational XCO works in the following way:

1. read a word from the input text.
2. check whether the word or any initial portion of it matches a member of the left-end exception list. If it does, exchange the word or the (initial) portion of it that matches the pattern with the corresponding exchange-string member of the exception list.
3. if the final portion of the word can be identified with an inflexional ending found in the list of endings, strip the ending off, i.e. identify the stem final letter.
4. check whether the stem or any final portion of it matches a member of the right-end exception list. If it does, exchange the stem, or the (final) portion of it that matches the exceptional pattern, with the

corresponding exchange-string member of the exception list.

5. ready to take next word.

C. COMPOUNDS

There can be no doubt that the existence of compounds of varying depth is a serious obstacle to constructing a fast and efficient XCO. At present it seems almost inevitable that some sort of morphemic analyzer along the lines of the one used in the MITALK text-to-speech system for American English (see Allen, 1981) must be integrated in the algorithm as a sort of preprocessor. This implies that the list of exceptions/exchange-strings must simply be a root-lexicon which has to be consulted for each word in the input text. If such a lexicon must be constructed anyway, one might as well include information concerning word class, syntactic behaviour, and the like, since such information will probably turn out to be indispensable to the correct prediction of syntactically determined stress reductions.

The inclusion of a root lexicon would also solve the problem of predicting stress placement in polysyllabic root morphs. Rischel's work in this field (Rischel, 1969) should of course be consulted, but it must be remembered that his rules are not concerned with orthography. For instance, in a root lexicon there would have to be information to the effect that the *æ* in *mausolæum* and the *o* in *petroleum* are long, since stress rules of the type set up by Rischel rely heavily upon the distinction between long and short vowels.

D. HOMOGRAPHS

Unfortunately, Danish orthography has rather an abundance of heterophonous homographs. In most cases such pairs of homographs consist of a noun and a non-noun, cf. *hul* [hɔl] 'hole' vs. *hul* [hu:'l] 'hollow', *bad* [bað] 'bath' vs. *bad* [ba:'ð] 'asked, prayed', and many others. The existence of homographs presents a difficulty which cannot be solved in any principled way in a text-to-speech algorithm as long as semantic information is beyond the scope of the system. It may be mentioned in passing, however, that the fact that one member of a pair of homographs is often a noun, whereas the other is not, makes it possible in certain cases to exploit the syntactic behaviour of different word classes and thus circumvent the lack of semantic information. E.g. *hul* can be resolved if it is preceded by the indefinite article *et*, in which case it must be the noun; but such devices are expensive (each one will cost one more "rule"), and they cannot, of course, be integrated in the algorithm in any principled way.

IV. CURRENT AND FUTURE WORK ON THE GTPA

Since the main difficulties with constructing a GTPA lie in the XCO, the current work is concentrated on extracting regularities from the Danish orthographic conventions.

This work is a sort of trial and error procedure. Tentative rules are stated on the basis of partial orthographic regularities. The consequences of such tentative rules are tested by feeding the operational GTPA program - which was developed during the spring and summer of 1982 and which runs on the PDP11/60 computer assigned to the text-to-speech project - with hopefully representative samples of running text. In each such test the result may either be that it is found worth while to elevate the provisional rule to a genuine rule, which in the majority of cases leads to the establishment of a corresponding set of exceptions; or the result may be that the rule is discarded.

The operational GTPA program can also be used to test phonological rules of Danish in much the same way as the DANFON project (cf. Basbøll and Kristensen, 1975).

The data on which the tentative rules are based is a Danish dictionary (Holmboe, 1978) which is stored on disc, and which can therefore be manipulated at will by means of text processing programs.

There is no doubt that, apart from the problems arising from the existence of homographs, the problems of deriving reliable allophonic representations from orthographic text can in principle be solved at the word level. Indeed, this goal seems within reach in the course of a few months. The integration of syntactically determined aspects of Danish pronunciation still awaits thorough research.

REFERENCES

Allen, J. 1981: "Linguistic-based algorithms offer practical text-to-speech systems", *Speech Technology 1,1*, p. 12-16

Basbøll, H. and Kristensen, K. 1975: "Further work on computer testing of a generative phonology of Danish", *Ann. Rep. Inst. Phon. Univ. Cph. 9*, p. 265-291

Holmboe, H. 1978: *Dansk Retrogradordbog*, (Akademisk forlag, Copenhagen)

Rischel, J. 1969: "Morpheme stress in Danish", *Ann. Rep. Inst. Phon. Univ. Cph. 4*, p. 111-144

Sherwood, B.A. 1981: "New technology provides computer voices for education", *Speech Technology 1,1*, p. 25-29.