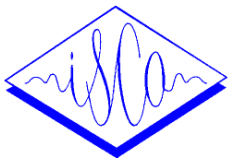

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

A WRITTEN TEXT PROCESSING EXPERT SYSTEM FOR TEXT TO PHONEME CONVERSION

Michel DIVAY

Institut Universitaire de Technologie, Université de Rennes,
BP 150, 22302 Lannion, France

ABSTRACT

Converting a word or a written text from characters to phonemes is one of the first steps done in text-to-speech synthesis. The first part of this process is a text normalization replacing numbers, abbreviations and acronyms, by their full text equivalents. The second step is grapheme to phoneme conversion.

Phonetics can also be used to search a file using a phonetic index when the spelling of the name (client file for example) or of the word (look-up in a dictionary) is not certain. In the last case, phonetics can be used to correct spelling mistakes in a text or to find a word in a dictionary without knowing how to write it.

A specialized programming language has been designed to write the rules. This allows the rules to be easily defined, revised and expanded independandly of the program (compiler and interpreter).

For French, the lastest version consists of approximately 500 rules and is used in different products (synthesizers, CDROM).

A different set of rules could be defined for proper names or for other languages: English, Spanish, etc.

I. INTRODUCTION

For both French and English, the text to phoneme transcription is complicated because, due to their Latin origin and evolution, a gap has slowly been created between what is written and what is pronounced. This presents a difficulty for the speech synthesizer as it does for young pupils or foreigners learning how to read the language [6].

A single phoneme can be produced by several letters (/i:/ is produced from the following underlined characters s<u>ea</u>, s<u>ee</u>, ac<u>e</u>tic, achi<u>e</u>ve, <u>ae</u>olian, ambergr<u>i</u>s, am<u>oe</u>ba, A.<u>D</u>., etc).

On the contrary, a single letter can be pronounced in different ways : "x" is pronounced [ks] in axiom, [gz] in example, [k/] in anxious and not pronounced in Bordeaux.

For proper names, the correspondance between written names and pronunciation is, due to their various origins, even more difficult to specify.

II. THE FORMALISM (THE SYSTEM)

Expert systems make a clear distinction between expert and programming (system) competence. We are all "expert" in reading our own native language, and we apply, albeit unconsciously, rules to read aloud a text.

Developing a program to automatically convert a text must be done in such an expert system manner. Thus, a specialized programming language has to be designed to express the rules. This language must be as compact, clear, and natural as possible.

External codes

First, the external codes of the units to be processed must be declared. For transcription, the graphemes (upper and lower case letters, numbers, punctuation, diacritics) and the phonemes codes have to be declared. These external codes may be composed of one or many characters. Some precautions must be taken to avoid ambiguity when defining the external codes.

These basic input and output units, or "elements", may be expressed as e_i.

Strings

A string consists of the concatenation of different elements. $e_1e_2e_3$ is a string.

Classes

A class is a set of strings having a common property.

'V' : a, e, i, o, u, y/

defines 'V' as the class of vowels.

'Prefix' : anti, auto, bio, macro, micro, mini/

represents the class of prefixes.

Blocks of rules

The language consists of one or several blocks of rules. Each block describes a process, taking the input text, processing it, and replacing it by the result of the processing.

```
begin {Block i}
      rule 1
      .....
      rule n
end
```

Rules

The syntax of a rule is :

<number> : <ls> --> <rs> / <lc> + <rc>;

number is the rule label.
ls (left string) is the string to be replaced.
rs (right string) is the string replacing **ls**.
lc (left context) represents the strings we must find on the left side of **ls**.
rc (right context) represents the strings we must find to the right of **ls**.

lc and **rc** are formed with operands (characters, strings, classes) and operators (concatenation, logical or, negation).

examples :

1 : xy --> [a][c] / 'V' + 'C1' . L ;

The string "xy" is replaced by the string [a][c] if
- an element of the class 'V' is found on the left of "xy",
- and an element of the class 'C1' followed by an L is found on the right of "xy".

2 : x --> [b] ;
"x" is always replaced by [b].
There is no context.

Replacement with one or two buffers

The **rs** string replaces the **ls** string. This can be done using only one buffer, for both input and output, but when writting the left context, we must take into account the modifications done when processing the input string from left to right.

With two buffers, the modifications are done in an output buffer and copied into the input buffer at the end of the block of rules. This means that in effect there are 3 contexts : the left and right context in the input string, and the left context in the output string. This left output context is written between angle brackets.

5 : b --> / <[b]> + ;
means b is eliminated if, on the left context of the output buffer, there is a [b].

Formal examples of rules

E = {A, B, F, G, H, X, Y,Z}; input and output characters
'C1', 'C2', 'C3' classes formed with strings of E.
'C1' : AG, HX/
'C2' : AX, BX, Y/
'C3' : HH, GG/

11 : A --> / A.'C1' + ;
A is replaced by nothing (eliminated) if, on the left side of A, an element of the class 'C1' is found preceded by another A.

12 : AB, CD --> XY / 'C1'.B + 'C2'.E, 'C3';
The string AB or CD is replaced by XY, in the following contexts:
- on the left of AB or CD, a B preceded by an element of the class 'C1',
- on the right of AB or CD,
  - either an element of 'C2' followed by an E,
  - or an element of 'C3'.

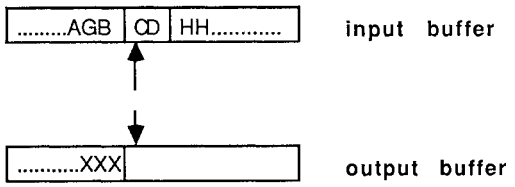

Fig.1 - Example of correct contexts for rule 12 (AG element of C1, HH element of C3).

13: A --> B / Non(F,G) + ;
A is replaced by B if the left context is not an F or an G.

14: XY --> AB / <G.'C1'> F +;
XY is replaced by AB if :
- the left context of the output buffer (between angle brackets) is an element of 'C1' preceded by G,
- and the left context (input buffer) is F.

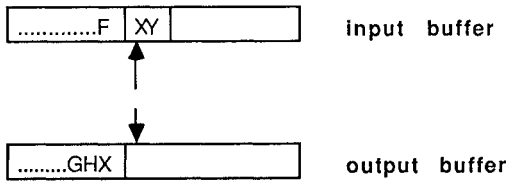

Fig.2 - Example of correct contexts for rule 14 (GH element of C1).

The rule interpreter

Rules are not tested sequentially. In a block, the rule having the "longest match" between its left string (**ls**) and the input string being processed is searched first. If both contexts are true, the rule applies, otherwise another rule is searched for, first any other rule with the same **ls**, and then in decreasing length of **ls** matches.

Let us consider the following rules:

```
begin
    100 : XY    --> ...;
    101 : X     --> ...;
    102 : XYZ   --> ...;
    103 : XY    --> ...;
    104 : YZ    --> ...;
    105 : XYZH  --> ...;
end
```

and the input string, "XYZF" to be processed.

The longest match between the left string (**ls**) of the rules in the block, and the input string to be processed is looked for. In this case the longest match is "XYZ". So, rule 102 is tested. If the contexts are true, the rule is applied, and the next character to process is "F" in the input string.

If the context is false, the rules are tested in decreasing order of longest match. Rules with "XY" as **ls** are tested in the order in which they are written (100, 103). Then if no rule has yet been applied, the rule 101 is tested. If no rule is true, the first character to process is copied (X in the example) into the output buffer, and the procedure starts again with the next character (Y in the example).

The order in which rules are tried is : 102, 100, 103, 101. The order in which the rules are written is significant only for those having the same **ls**.

III. USING THE FORMALISM TO DEFINE THE RULES OF THE EXPERT

Several examples will be given for both French and English grapheme to phoneme normalization and transcription.

Normalization consists of replacing numbers and abbreviations by their full text equivalents. It consists of one or several blocks of rules preceding the transcription.

Examples of rules for French

Normalization (for French)

1) H should be replaced by "heure" if it is a duration as in 132H15, but not as in H. Ford.

```
'Digit' : 0,1, 2, 3, ..., 9/   is the class for digits
230 : H,h  --> heure / 'Digit'  + 'Digit', _;
```

H or h is replaced by "heure" if the left context is a digit (element of 'Digit'), and if the right context is a digit or a space (underline means a space)

In the example, 132H15, "H" would be replaced by "heure".

2) If we are only working with times in 24 hour clock format eg. 9H, 9H30, 15h25, 23H05, 12H, the rules may be written in a more restrictive way:

```
'D04': 0, 1, ..., 4/   digits between 0 and 4
'D05': 0, 1, ..., 5/   digits between 0 and 5

240 : H, h --> heure / _.'Digit', _.1.'Digit', _.2.'D04'
          + 'D05'.'Digit'._, _;
```

3) 250 : mm --> millimètre / _.'Digit' + _;

mm is replaced by millimètre in "35mm" or "trois mm", but not in "grammaire".

4) Similar rules are used to spell the acronyms (I.B.M gives I bé ème) or to replace numbers by their equivalents in letters.

277 is rewritten as "deux cent soixante dix sept" by a set of rules checking the left and right context for each digit.

```
260 : 2 --> deux cent / + 'Digit'.'Digit'._;
```

2 is rewritten "deux cent" if followed (right context) by two digits and a space.

```
270 : 2 --> deux / + _;
```

2 is rewritten "deux" if followed by a space.

Transcription (for French)

5) Usually, as a general rule, "s" is pronounced [s] as in saint, salon, sérum, sofa, absent, seconde.

There are exceptions to this general rule ."s" between two vowels is pronounced [z] like in basalt, vase, maison.

But there are exceptions to these exceptions. If "s" follows a prefix, it produces an [s] as in antisocial, autosuggestion, microseconde, parasol.

The rules could be written as shown below.

```
'Prefix' : anti, auto, micro, para/      Class of prefixes

1100 : s    --> [s] /     _.'Prefix' + 'Vowels';
1101 : s    --> [z] /     'Vowels' + 'Vowels';
1102 : s    --> [s];
```

6) In French, "c" is pronounced [s] if followed by e, i, é, è, ê, î as in centre, cigarette, accès; otherwise it is pronounced [k] as in case, cobra.

"g" gives [j] before the same vowels (agent, agile) otherwise [g] (algorithme, garage).

The rules are the following:

```
'VACG' : e, i, é, è, ê, î/    Vowels after C ou G

1200 : c --> [s] /  + 'VACG';
1201 : g --> [j] /  + 'VACG';
```

7) The "ai" string in the French words bienfaisant, contrefaisait, défaisait, faisan, satisfaisant, etc, is pronounced [e] but not in faisceau, chauffais where the corresponding phoneme is an [è].

The rule can be written as :

```
1300 : fais --> [f][e][z] / + 'Vowels';
```

"fais" is pronounced [fez] if "fais" is followed by an element of the class 'Vowels'.

```
8) 1400 : c, k, qu  -->    / <[k]> + ;
```

"c", "k", "qu" are eliminated if, in the output buffer, there is a phoneme [k]. This would be the case for the second "c" of accaparer, the "k" of cricket, or the "qu" of grecque, the first "c" having produced a [k] in the output buffer.

9) A similar set of rules is used to deal with the problem of linking between two words. A silent consonant at the end of a word can be pronounced or not depending on the following word as in "nous avons". If there is a linking, the "s" of "nous" gives a phoneme [z].

The mute "e" is often elided at the end of a word ("grande"), and depending on the speech rate, in the middle of words preceded by one consonant ("dangereux") but not if there are two consonants before ("premier"). Elision is possible, in the first syllable ("religieux") but is considered bad style, and never done for words like "bachelier".

Examples of rules for English

The system has not been used for English, except for a few tests concerning the consonants. Some examples of potential uses are given for people not familiar with French (previous paragraph).

1) 2010 : cc --> [k][s] / 'Vowels' + e,i;

This rule means that the string "cc" is replaced by [k][s] if "cc" is preceded by a vowel, and followed by "e" or "i", as in accept, access, succeed, accident, succint but not in occasion, account, accuse.

2) 2011 : c, k, qu --> / <[k]> + ;

c, k, qu are deleted (replaced by nothing) if the left context of the output buffer is [k].

This works for

- the second "c" of according, occasion, raccoon,
- "k" in back, checking, cricket,
- "qu" in acquire, racquet.

3) 'CeH' is the class of consonants excluding H.

2012 : c --> [k] / + 'CeH';

works for

- "c" in anecdote, back, clean, cream, graphics
- the first "c" of according, occasion, raccoon

but not for "c" in achieve or attach

4) For instance, the translation of "cc" in "occasion" is performed as indicated below.

Rule 2010 is tested but does not apply: the right context is false.

Rule 2011 is tested with the first "c" but does not apply: the left context of the output buffer is not [k].

Rule 2012 applies for the first "c", so a [k] is sent to the output buffer, and the next character to process is the second "c".

Rule 2011 applies for the second "c", thus deleting this second "c". The next character to process is the "o" following "cc", and so on until the last character of the word.

Examples for parsing

The formalism has been used to help parsing a sentence. After a dictionary look-up, a set of possible grammatical categories is attached to each word. By examining the grammatical categories of the words on the left and on the right, it is possible to reduce (to one if possible) the set of potential grammatical categories for each word of a sentence. The same formalism allows the processing of strings of grammatical categories instead of characters for the transcription. A class is a set of grammatical categories [2][3].

IV. APPLICATIONS

This program is currently in use in different laboratories in France, Canada [1] and USA as the first step of speech synthesis for French. The formalism could be easily used for other languages.

It is used by various companies producing electronic board speech synthesizers for French, and in the French Dectalk prototype (Digital, Maynard, USA).

This transcription program has also been used to create a phonetic index and retrieve a word without knowing how to write it. The word is converted to phonetics and searched for in the phonetic dictionary index (used in the CDROM dictionary "Le Robert Electronique") [5], giving the correct spelling or the word definition. The same mechanism is used to retrieve proper names (of employees or clients) without knowing how to spell them.

The system for French consists, in the most recent version, of 500 rules and 80 classes, some of them composed of a hundred or more strings. It has been systematically tested on a 100,000 word dictionary ("Le Grand Robert de la Langue Française"). An exception dictionary of approximately 6000 words has been created for words not correctly translated by these 500 rules, consisting mostly of foreign words such as accelerando, jeans, mea culpa, steward.

V. BIBLIOGRAPHIE

[1] D. O'Shaughnessy, M. Lennig, P. Mermelstein, M. Divay, "Simulation d'un lecteur automatique du français", 12 ème JEP (Journées d'études sur la parole), Mai 1981, Montréal, Canada.

[2] M. Divay, "De l'écrit vers l'oral ou contribution à l'étude des traitements des textes écrits en vue de leur prononciation sur synthétiseur de parole", Thèse d'Etat, 1984, Université de Rennes I, France.

[3] M. Divay, "A text-processing expert system", 5ème Congrès Reconnaissance des formes et Intelligence Artificielle, Novembre 1985, Grenoble, France.

[4] E. Laporte, "Méthodes algorithmiques et lexicales de phonétisation de textes", Thèse, Université Paris 7, Mai 1988

[5] A. Rey, A. Duval, B. Vienne, B. Struyf, M. Divay, T. Lootens, S. Zimmermann, "Le Robert Electronique, ensemble d'outils d'aide à la rédaction de textes français sur disque optique compact (CDROM)", Novembre 1989.

[6] M. Divay, "Traitement du langage naturel : la phonétisation ou Comment apprendre à l'ordinateur à lire un texte français", MICRO-SYSTEMES, Mars 1990.