Optimising Tools for the French Letter-to-Phone Grammar TOPH With a View to Phonographic Spelling Correction

Nada GHNEIM & Véronique AUBERGÉ
Institut de la Communication Parlée
INPG/Université Stendhal, URA CNRS n° 386
BP 25, 38040 Grenoble Cedex 9, FRANCE
Phone: (+33) 76 82 43 38 Fax: (+33) 76 82 43 35
e-mail: ghneim@icp.grenet.fr, auberge@icp.grenet.fr

Abstract
The goal ,in the long term, of this work is to give a formal and linguistic description of the text-to-phone processing, and to use the description in a view to phonographic spelling correction errors. This description can be formulated, by TOPH language, in a determinist grammar.

The French letter-to-phone exhaustive grammar *TOPH* is the base of our work in phonographic spelling correction: the inverse grammar *PHOT* gives the phone-to-letter correspondences. However, as it was constructed by human expert, *TOPH* has many redundant and incoherent rules which affect correction results. By eliminating these rules we got an optimal grammar to be used in the spelling correction.

We have developed an environment around a French letter-to-phone system, which enables to filter the redundancies and the incoherences in a lexical grammar written in *TOPH* language. The result is an optimal grammar in its logic and linguistic description. Such an environment had never been developed for a French letter-to-phone system. We use this exhaustive phonetic description to establish the duality between letter-to-phone processing and phone-to-letter one, with a view to phonographic spelling correction.

I. Introduction

The right use of spelling is a crucial problem in French language: on the one hand, the spelling presents real difficulties [Catach 89], and on the other hand, a good skill of spelling is an important social criterion. This is why automatic error correction of French text was the object of many studies [Pérennou 86, Laporte 89, Strube de Lima 90, Véronis 93].

In this work we are interested in the phonographic spelling correction errors, in which the writer substitutes a phonetically "close" but orthographically incorrect sequence of letters for the intented words.

A minimal French letter-to-phone grammar is the core of an actual grammar, extended following a systematic methodology in exploring a French representative dictionary: the ICP's[1] dictionary *"Le 60000"*, and in referring to the Petit Robert1[90][2] for the phonetic entries [Belrhali 92].

Rules order is an essential parameter in *TOPH* grammar description: expert expresses his knowledge naturally with the logic of exception followed by a general case, which means: "If a grapheme (*i.e.* a sequence of letters correspondent to an unit of pronunciation) is in a singular context, then it follows a singular transcription; Else, it follows the usual transcription"

[1]Institut de la Communication Parlée

[2]Commun French dictionary with 80.000 entries

When adding a new rule, the expert must take into account the meaning of this rule induced by the rules order, and by their syntagmatic concatenation on the word. As the grammar complexity increases, the control of the implicit logic becomes more difficult. Then it is necessary to treat non optimal grammars, and overall to detect prospective rules redundancies and incoherences, which do not affect, or not much, the text-to-phone global results: grammar determinism is ensured by rules writing order. On the other hand, word treatment time, and necessary memory size are uselessly increased, and the linguistic description becomes non optimal. Moreover, if we consider the PHOT phone-to-letter grammars, which is the result of the inversion of *TOPH* grammars, parasite phones introduction, by redundant or incoherent rules, makes invalid the linguistic formality of *PHOT* grammars.

In the next two paragraphs we introduce *TOPH* and *PHOT* systems, after that we present different tools to verify the coherence and the redundancy of language *TOPH* rules.

2. *TOPH* system

TOPH (Transcription from Orthograph to PHonetics) is a multilanguage text-to-phone system, which allows to transliterate a graphic string to a phonetic one.

A *TOPH* language grammar is constituted of two modules:

1. The first module contains the declaration of the sets, which allows to describe in the same syntax:
   - Phonetic class, like the set of nasal consonants: "Nasal Consonants" = (n, m, ...)
   - Lexicons of words which correspond to non regular etymological families, for example the lexicon of words in which the letter "s" is pronounced in final, which is not generally the case in French where a mute "s" is a plural mark: "Lexicon s final" = (bu, consensu,...), which corresponds to the set (bus, consensus,...)
2. In the second module transposition rules (partitioned into classes) are described. Rules class is determined by the first character in the transliterated string. Rule syntax has the following form:

(CtxtG) +Ch_graph+ (CtxtD) {Ens_Ctg} —> [Ch_phon]

where CtxtG, CtxtD are respectively left and right contexts which could be empty, Ch_graph is the string to transliterate, Ens_Ctg is the set of lexical categories with which this rule should be applied (could be empty in general when the rule does not treat a morpho-phonologic or morpho-syntactic ambiguity), Ch_phon is the correspondent phonetic string.

a. Organisation of a class: vertical order

Rules in the same class are not defined independently one of others; rules complete contexts are not explicitly given, but they are described by a local order relation, which is defined as "vertical", on the class of rules. This order is the one given sequentially by the expert, for example:

R1: ("Vowel") +s+ ("Vowel") —> [z]

R2: +s+ —> [s]

If R1 is a rule followed by R2 (R1, R2 means that the grapheme "s" in a vocalic context is voiced and pronounced [z] instead of [s] in the general case),then the explicit description of R2 is R3:

R3: (α) +s+ (β) —> [s]

where $\alpha, \beta = V_T \backslash$("Vowel"), where V_T is the terminal vocabulary of the grammar.

The expert develops rules in each class with the logic "If R1 is applicable then apply R1; else if R2 is applicable then apply R2; else ...". So, in each class, the grammar is always determinist, even if the complexity of a class makes the administration of the vertical order (by the expert) very difficult.

b. Organisation of the grammar: horizontal order

We have seen that the grammar determinism is ensured, inside the same class, by rules vertical order. The determinism of the grammar all over the classes is ensured by another mechanism of order in rules application, which will be defined as an "horizontal order" on the grammar. To define this order, let the string "gu" be the input string,:

Extract of "gu" class

R4: +gu+ —> [g]

R5: +g+ —> [g]

If we explicit R5 according to intra-classes vertical order, R4 and R5 will be independent and stated as:

R4: +gu+ —> [g]

R5: +g+ $(V_T \backslash u)$ —> [g]

The two rules R4 and R6 describe the same transcription window "gu":

R6: (g) +u+ —> [u]

R4 transliterates all "gu", R6 transliterates "u" in "gu". So, there is an ambiguity. However, input order is imposed from left to right; which implies that R4 will be applied, and R6 will never be taken into account (R6 is inaccessible).

Then, input order (from left to right) implies, implicitly, an horizontal order of inter-classes rules application.

Horizontal order sets the expert the fundamental problem of the choice of graphemes: it is difficult to maintain the coherence of this choice between classes, because rules are developed class by class according to the vertical order.

3. *PHOT* System

PHOT (Transcription from PHonetics to Orthograph) is a description language for spelling grammars. *PHOT* grammar rules are obtained by the inversion of *TOPH* grammar, grouped in phonetic classes. A *PHOT* grammar is composed of two modules:

1. The first one contains the declaration of the sets (the same of the initial grammar *TOPH*)

2. The second contains transcription rules which are partitioned in classes of phones. A rule syntax has the following form:

(Ch_graph) —> (CtxtG) +[Ch_phon]+ (CtxtD) {Ens_Ctg}

Example:

R7: [s] —> ("#"+tourne) +s+ (ol+"#")

R8: [z] —> ("Vowel") +s+ ("Vowel")

Then, "s" in a vocalic context, in the word "tournesol" (turnsole) [tuʀnəsɔl], is not pronounced [z] but [s] to give the oral information of the morphologic structure, which is the agglutination of the two lexemes "tourne" (to turn) and "sol" (sun).

This grammar, which describes correspondences between a sequence of phones and the set of contextual strings which could produce it, will be the base of our model for the phonographic spelling correction.

4. Preliminary definitions

- A rule is called *redundant* if there are other rules in the grammar, which treat all (or a part) of the same graphic string (with correspondent contexts), and generate the same phonetic transcription.
- A rule is called *incoherent* if there are other preceding rules in the grammar, which treat all (or a part) of the same graphic string (with correspondent contexts), and generate the same phonetic transcription.
- *The local (or global) study of the redundancy* of a given rule consist in looking for the first (or all) redundant rule(s) in the grammar.
- *The local (or global) study of the incoherence* of a given rule consist in looking for the first (or all) incoherent rule(s) in the grammar.

5. Utility of the research and treatment of redundancy and incoherence

The treatment of redundant and incoherent rules is very important to optimise memory and computation time, and also from a rules linguistic description validity point of view :

5.1. Redundancy and incoherence effect on phonetic transcription execution time

The necessary time to transliterate a word is equal to $\sum_{gr \in word} T(gr)$, where $T(gr)$ is the necessary time to find the applicable rule on the graphic string gr in the correspondent class.

The average time necessary to find the rule is: $T_{average}(gr) = \sum_{i \in [1..n]} T_i(gr) \times \Pr op_i$

where n is the number of rules in the class correspondent to the string "gr", $T_i(gr)$ is the necessary time to reach the ith rule (equal to i operations) and $\Pr op_i$ is the probability of using the ith rule.

In *TOPH*, the last rule of a class is always the general one (supposed to be the most frequently used). So, in this case:

$$T_{average}(gr) = \sum_{i\in[1..n]} i*\Pr op_i \geq \frac{n+1}{2} operations$$

Therefore, the time necessary to transliterate a word depends on the number of grammar and on the position of each rule inside the class.

5.2. Redundancy and incoherence effect on the size of memory

The necessary memory to store rules is:

$$Memory_{total} = nb_{rules} * Memory_{rule}$$

where nb_{rules} is the number of rules in the grammar, and $Memory_{rule}$ is the constant memory size necessary to store one rule.

5.3. Redundancy and incoherence effect in determining the set of phones in *PHOT*

We have seen that it is difficult, for the expert, to determine, in a single way, the graphic string which he must choose as string to transliterate in *TOPH* rules. While *TOPH* grammar determinism is ensured by the horizontal and vertical order, this choice could modify the set of phones, which implies considerable modifications in *PHOT*.

Example

Considering the following rules in *TOPH* grammar:

R9 :	("#"+qu) +ia+ ("#") —> [ija]	("quia" —>[kija] (*quia*))
R10 :	("#"+qu) +i+ (é,a) —> [ij]	("quiétude" —> [kijetyd] (*quietude*))
R11 :	+a+ —> [a]	

When inversing this grammar, we will have the strings "ija", "ij" and "a" (correspondent to the graphemic strings "ia", "i" and "a") in the phonetic input group. Then, in eliminating R9, which is redundant comparing to R10 and R11 in *TOPH*, the phonetic string "ija" will be, consequently, eliminated from the set of phones concerning *PHOT*.

6. Local study of the redundancy and the incoherence

The local check of redundancy and incoherence is applied on a new inserted rule, in an optimal grammar, in which there is neither redundancy nor incoherence. In this case the research is stopped when finding the first rule which can be applied on the string to transliterate of the new rule.

6. 1. general method

To test redundancies and incoherences of rules in the grammar, we proceed as follows:

- Remove the studied rule from the grammar to get the new grammar.
- Transliterate the grapheme of the studied rule, considering right and left contexts, using the new grammar.
- If there was no applicable rule in the new grammar then the studied rule is not redundant neither incoherent.
- Else, compare the result of the transcription (the phone produced by the applicable rule) with the phone produced by the studied rule:
  – if they were equal then the studied rule is redundant to the applicable one.
  – else the studied rule is incoherent to the applicable one.

6.2. Computational time

If we explicit the j[th] rule under the form:

$$\left(Set_{0j} + Set_{1j} + \ldots + Set_{ij} + \ldots\right) + Ch\_graph + \left(\ldots + Set_{lj} + \ldots + Set_{NbConcat_j}\right)\{Ens\_Ctg\} \rightarrow [Ch\_phon]$$

where Set_{ij} is the set of strings limited by two plus signs, in the j[th] rule. Then, total execution time is of the order: $\sum_{j=1}^{NbRules} \prod_{i=1}^{NbConcat_j} \left|Set_{ij}\right|$, where $\left|Set_{ij}\right|$ is the cardinal of the set i, in the j[th] rule.

6.3. Memory size

Memory size is proportional to the number of rules in the grammar:

$$Memory_{total} = nb_{rules} * Memory_{rule}$$

6.4. Examples of local redundancy

Rules are redundant if :

- They are identical in their expression.
- They can produce the same string

R12 : ("#"+"EXCEP:OO") +oo+ —> [u] (*e.g.* "booling" —> [bulin] (*booling*))

R13 : ("#"+igl) +oo+ —> [u]

where the set "EXCEP:OO" contains the element "igl".

- One string produced by a rule is a sub-string of a string which is produced by another one. For example (the following rules treat etymologically Greek words):

R14 : (spiro, sti, sto, stœ, syn, sporotri) +ch+ (a,è,é,i,o) —> [k]

R15 : (tri) +ch+ (i) —> [k]

where "trichi" ⊂ "sporotrichi", and "trichi" is a string produced by R15 and "sporotrichi" is an element of the set of strings produced by R14.

• One rule can be replaced by a set of other general rules, for example with the following rules we can have the same phonetic transcription obtained by the rule R16, in applying successively R17 and R18:

R16 : +iu+ (m+"#") —> [jo] (Latin final "-ium", *e.g.* "atrium" —> [atrijɔm])

R17 : +i+ ("Vowel") —> [j] (general rule, like in "sioux" —> [sju] (*Sioux*))

R18 : +u+ (m+"#") —> [o] (Latin final "-um", *e.g.* "quantum" —> [kwɑ̃tɔm])

6.5. Examples of local incoherence

Incoherences occur when:

• The expert puts a rule treating a particular case after the general rule, so the system stops always before reaching the second, for example :

R19 : ("#"+st) +ea+ (k+"#") —> [ɛ]

R20 : ("#"+st,str,sw) +ea+ —> [i] (like in "steamer" —> [stimœʀ] (*steamer*))

• The expert treats the same case using graphemes with different length, for example:

R21 : ("#"+séqu) +o+ (ia) —> [ɔ] ("séquoia" —> [sekɔja] (*sequoia*))

R22 : ("#"+séqu) +oia+ ("#") —> [ɔja]

• The expert treats the same case in two different ways :

R23 : ("#"+m) +oe+ (re) —> [] ("moere" —> [mwɛʀ] (*polder*))

R24 : (m) +oe+ (re) —> [wɛ]

7. Global study of the redundancy and the incoherence

In this case, the rules of the grammar are written by the expert without local treatment of the redundancy or the incoherence. To optimise this grammar, the local research method explained before is incomplete: the research is stopped at the first applicable rule, and this will possibly hides other redundant or incoherent rules.

A global research method allows to identify all redundant and incoherent rules in the grammar, in extending the local research algorithm to find all the applicable rules on the string to transliterate.

7.1. Examples of global redundancy

• Redundancy global research of the studied rule detects one (or a set) of redundant rule(s), for example the rule:

R25 : ("#"+tr) +ou+ (ée+"#") —>[u] ("trouée" —> [tʀue] (*breach*))

has the following set of redundant rules:

R26 : (t,f,c,b,p+"Liquid Cons.") +ou+ —> [u]

R27 : ("#"+ encr,tr) +ou+ (é, er) —> [u]

R28 : ("Cons."+"Liquid Cons.") +ou+ ("Vowel") —> [u]

In this case, we could have rules which treat the same case (with contexts of different length), and then we could unify these rules, if possible, in one general rule.

• Detected global redundant rules are parasite: this case is the result of the exhaustive research of applicable rules, and it happens when a succession of letters implies the application of a general rule concerning this succession, but not in words in which another (more general) rule must be applied.
For example, the general rule R29 to transliterate the grapheme "a" in the succession ay"Vowel" is:

R29 : +a+ (y+"Vowel") —> [ɛ] (like in "payer" —> [pɛje] (*to pay*))

On the other hand, there are certain words like "pagaye" (disorder) which do not follow this rule but the general rule R30 of the class " a" :

R30 : +a+ —> [a]

but, because rules in each class are treated by vertical order, the expert must put the rule R31 which treats this word before R29:

R31 : ("#"+pag) +a+ (ye) —> [a] ("pagaye" —> [pagaj] (*disorder*))

This is why, the application of the research global of R31 redundancy will give R30 as a redundant rule (parasite redundancy). This result allows to find exception rules treating exception words, to put them in a set of exceptions "EXCEP:AY", and to replace R29 by R32:

R32: +a+ (y+"Vowel") \ ("EXCEP: AY") —> [ɛ]

where '\' means the "except" operation.

7.2. Examples of global incoherence

• Global research detect for the studied rule a set of incoherent rules, for example the rule:

R33 : +o+ (in+"Non Nas.Cons.") —> [w] (like in "lointaine" —> [lwɛ̃tɛn] (*distant*))

has the following set of global incoherent rules:

R34 : +oin+ ("#","Cons.") —> [wɛ̃]

R35 : +oi+ —> [wa]

• Detected global incoherent rules are parasite: this happens when a graphic string follows a certain rule, except in words (which must be given in rules situated before this rule, "vertical order" of applying the rules). Then the global research will detect the exception rules as incoherent to the studied one. For example, the general rule applied on the grapheme "u" followed by a vowel is R36:

R36 : +u+ ("Vowel") —> [ɥ] (like in "quidam" —> [kɥidam] (*person*))

but there are exceptions where this rule is not applicable, as in following rules:

R37 : +u+ (e+"#") —> [y] (like in "rue" —>[ʀy] (*street*))

R38 : +u+ (y) —> [ɥi] (like in "tuyauter" —> [tɥijote] (*to quill*))

which must be met before R36, and so detected as incoherent rule.

8. Conclusion

Redundant and incoherent rules, which are detected by the algorithm are not automatically filtered: in treating a *TOPH* French grammar of 1200 rules, we have established the fact that they reveal more much deep linguistic incoherence problems (usually in relation with the horizontal order, from which maximum graphemes are defined).

The list of rules is proposed to the expert, who has the choice to handle the grammar.

After obtaining the new optimal grammar *TOPH*, we inversed it to produce the correspondent grammar *PHOT*. In the *PHOT* system, the graphemic strings are generated under the strong constraint of right and left phonetic contexts (calculated like the phonetic correspondences of the orthographic contexts in the *TOPH* grammar) surrounding the processed phonetic string.

This is not the case of all the phone-to-letter systems developed for French spelling errors environments, which use correspondences between graphemes and phonetic strings without any contextual constraints. This is why the number of orthographic solutions produced for one phonetic string in such systems is very numerous in comparison with that produced using *PHOT*.

7. References

[Aubergé 91] V. Aubergé, "La synthèse de la parole: des règles aux lexiques", Thèse de l'université Pierre Mendès France, Grenoble2, 1991.

[Belrhali 92] R. Belrhali, L. Libert, V. Aubergé, L.J. Boë, "Élaboration des lexiques d'une grammaire de phonétisation du français", 19es JEP-SFA, Bruxelles, 1992.

[Catach 89] N. Catach, "Les délires de l'orthographe", Plon, 1989.

[Laporte 89] E. Laporte, M. Silberztein, "Vérification et correction orthographique assistées par ordinateur", Actes de la 1ère conférence européenne sur les techniques et les applications de l'Intelligence Artificielle en milieu industriel et de service, Hermès, 1989.

[Pérennou 86] G. Pérennou, P. Daubeze et F. Lahens, "Vérification et correction automatique de textes, prise en compte de fautes orthographiques et typographiques, Un modèle VORTEX", TSI, vol. 5, n°4, juillet-août, 1986.

[Strube de Lima 90] V. L. Strube de Lima, "Contribution à l'étude du traitement des erreurs au niveau lexico-syntaxique dans un texte écrit en français", Thèse de l'Université Joseph Fourier, Grenoble I, 1990.

[Véronis 93] J. Véronis, "Distance entre chaînes : extension aux erreurs phono-graphiques", Travaux de l'Institut de Phonétique d'Aix, vol. 15, pp.217-234, 1993.