Morphosyntactic correction in natural language interfaces

Jean VERONIS *
Groupe Représentation et Traitement des Connaissances
Centre National de la Recherche Scientifique
31, ch. Joseph Aiguier
13402 MARSEILLE CEDEX 9 - FRANCE

Abstract

Morphosyntax cannot be simply ignored in natural-language man-machine dialogue since it constitutes an important part of the meaning. Nevertheless, troublesome side effects can arise when morphosyntactic errors are combined with other types of errors. We describe here an efficient means of handling quite complex combinations of typographical, phonographic and agreement errors in French, which are typical of C.A.I. users : a sentence as erroneous as *les cotté adgassan à l'ippeauttainuz son perpndiqulère* (!) will be perfectly recognized and translated into *les côtés adjacents à l'hypoténuse sont perpendiculaires* (the legs adjacent to the hypotenuse are perpendicular).

I. Introduction

This study was carried out within the context of a C.A.I. system for teaching plane geometry at high school level, which is being developed at G.R.T.C (Chouraqui, Inghilterra and Véronis, 1988). In this system, natural-language interfaces occur in various places : experts are enabled to transfer knowledge (theorems, problems), and students to make demonstrations, using natural language. Error correction is particularly important in C.A.I. systems, since students are generally poor spellers and poor grammarians, and they make many conceptual errors in the subject they are learning.

We introduce a distinction between competence and performance errors. Performance errors are simply due to mechanical or neuro-motor problems (typographical errors, 'slips of the pen'), whereas competence errors reflect ignorance about language rules or misconceptions about the domain. Phonographic errors (in French : *ippeauttainuz* for *hypoténuse*) or agreement errors (*les côté opposé* for *les côtés opposés*) are typical competence errors. In man-machine communication, the correction of competence errors is far more important than the correction of performance ones (see Véronis, 1988c). In fact, when faced with an error message, the user can correct typographical errors, for example, but he will generally be unable to correct phonographic or agreement errors. He can only try various spellings at random, which is a rather frustrating way of interacting with a system. We have tried elsewhere (Véronis, 1987b, c) to demonstrate how some semantic and conceptual errors can be handled (especially wrong presuppositions) using a special many-sorted logic. The present paper focuses on morphosyntactic errors, inflexion and agreements.

We would stress that morphosyntax cannot be simply ignored in natural-language man-machine dialogue. Let us take, for example, the following wrong sentence, concerning a right triangle (we translate word for word; in French, determiners and adjectives agree in gender and number with nouns) :

Trace	*les*	*côté*	*opposé*	*à l'angle droit.*
(Draw	the$_{pl.}$	side$_{sg.}$	opposite$_{sg.}$	[to] the right angle).

Two corrections of this sentence can be performed :

Trace le côté opposé à l'angle droit (singular).
Trace les côtés opposés à l'angle droit (plural).

In the first case, there is no conceptual error, whereas, in the second, the user could (for example) have confused *côté opposé* (opposite side : there is exactly *one* such side, the hypotenuse) and *côtés adjacents* (legs adjacent to the right angle : there are two of them). This second interpretation

* The author's paper entitled "Une extension à la distance entre chaînes" was accepted at the COLING'86 conference in Bonn, and actually presented in the session Morphology. Due to some technical error the paper was not included in the final program and was omitted from the Proceedings.

should trigger an error message such as :

> *Warning : in a right triangle, there is exactly* ***one*** *side opposite the right angle, the* ***hypotenuse****. Do you want to see the figure (y/n)?*

We must therefore correct morphosyntactic errors (gender and number, but also person, tense and moods) with great care, and apply appropriate rules to find out the right interpretations.

The problem becomes rather more complicated when several types of errors (typographical, phonographic and morphosyntactic) are combined in a single word. Troublesome side-effects can then arise when a morphological program attempts to reduce such words to their root form. For example, the wrong form *hippothénuses* will be reduced to a hypothetical root form *hippothénuse,* which is not to be found in the dictionary. In addition, the inflexion itself may be misspelt (e.g. *démontron,* instead of *démontrons*). In such a case, the wrong ending may in addition no longer be a possible inflexion, so that the standard morphological program will fail in trying to construct a hypothetical root form. We therefore need a two-stage process, in order to first find out the root and inflexion of inflected words despite typographical or phonographic errors, and then to apply appropriate rules to obtain the right agreement interpretations. These rules will involve some weighting of the possible agreement errors, which makes certain interpretations more likely than others.

II. Root and inflexion retrieval

The most common strategy in spelling correction consists of applying reverse morphological transformations on words to produce a hypothetical root form, and then looking it up in the dictionary. If there is no matching entry, a spelling correction program is triggered. Nevertheless, if the inflexion is misspelt, the problem is really troublesome since, as mentioned above, the morphological program will be unable to produce a hypothetical root. The solution consisting of avoiding any morphological analysis by storing all inflected forms in the dictionary is a very inefficient one, since spelling correction algorithms all involve scanning a sometimes quite large portion of the dictionary. The time spent on spelling correction will then naturally be even greater in an inflected dictionary (remember, for example, that French verbs have about forty different inflected forms).

Moreover, much research has been devoted to spelling correction since the very beginning of computer science (for a review, see Peterson, 1980, and Pollock, 1982), but has generally focused on *noise* errors (due to hardware problems, such as errors caused by the input devices, or transmission) or *typographical* errors, due to keyboard typing slips, such as those listed in Damerau's (1964) often-quoted study, which shows that 80% of errors in words belong to one of the following categories :

- substitution of a letter for another,
- addition of a letter,
- deletion of a letter,
- transposition of two adjacent letters.

The first three errors can result from either noise or typographical causes, and the fourth is specifically a typographical one. We agree with Damerau (1964) that when writing computer programs or indexing documents by means of keywords, these errors are almost the only ones which occur. The same words are constantly repeated, and the operator (a specialist) knows exactly how to spell them. The mistakes made are therefore nearly all *performance* errors. But when the general public (especially in C.A.I.) uses computer services, very different problems can arise. Performance errors are still present, of course, but they are coupled with a very large number of *competence* errors such as phonographic ones, which, as we said previously, must be dealt with first and foremost.

The mathematical framework developed for noise and typographical errors is very badly suited to phonographic errors. For example taking Wagner and Ficher's (1974) and Lowrance and Wagner's (1975) distance between strings (based on *edit operations* which model Damerau's four kinds of errors), the wrong spelling *ippeauttainnuz* is very far from the right one *hypoténuse,* though it is obvious to any French speaker that the pronunciation is exactly the same. In addition, methods based on a transcription of words into some phonetic form cannot work when phonographic errors are combined with typographical ones.

We have therefore extended the notion of proximity between strings to take phonetic similarity into account. In the case of phonographic errors, a whole *grapheme,* which can be more than one letter long, can be replaced by another grapheme having the same phonetic value. This defines a similarity relation between graphemes, as shown in Figure 1. The basic idea is to extend the edit operations to similar-substring substitution, and to associate *high costs* with edit operations altering pronunciation (most noise and typographical errors) and *low costs* to edit operations preserving pronunciation (phonographic errors) (Véronis, 1988a, b).

In addition, we established a precise quantitative inventory of sound-to-spelling correspondences, which, although absolutely necessary in any attempt to build efficient phonographic correctors, was sorely lacking for French. This collection of data has subsequently proved to

be useful to both psycholinguists and teachers (Véronis, 1986, 1988d).

	c	ch	cqu	q	qu	k	s	ss	z
c	X	X	X	X	X	X	X	X	
ch	X	X	X	X	X	X			
cqu	X	X	X	X	X	X			
q	X	X	X	X	X	X			
qu	X	X	X	X	X	X			
k	X	X	X	X	X	X			
s	X						X	X	X
ss	X						X	X	
z							X		X

Figure 1 : part of the similarity relation between substrings with French

This led us to the building of an efficient algorithm for retrieving from a dictionary words which can be riddled by both phonographic and typographical errors. This algorithm is an extension to phonographic errors of the algorithm proposed by Damerau (1964), Morgan (1970), and Durham *et al.* (1983). There are two essential differences between the latter and the algorithm that we propose. First, we try to match the entire unknown word against a dictionary of *root forms,* as we shall describe later. Secondly, we scan the strings x and y from left to right, no longer by simply checking at each point (i, j) that the symbols $x[i]$ and $y[j]$ are the same, but rather by testing whether these symbols constitute the beginning of any *similarly-pronounced substrings.*

1) As long as $x[i]$ and $y[j]$ are the beginning of similar substrings, the indexes i and j are incremented by the lengths of the respective similarly-pronounced substrings, and this step is repeated (Fig 2.a).

2) When two symbols are found which do not fulfill this requirement (Fig. 2.b), the following four hypotheses are tested (they correspond to typographical errors) :

- the next two adjacent letters have been transposed,
- the next letter is missing (as in the example),
- the next letter has been inserted,
- the next letter has been replaced by another.

In each case, it is attempted to match the tail substrings according to 1), while skipping the appropriate letters (Fig. 2.c).

3) When the hypothetical root form has been completely scanned, if some substring remains in the unknown word, it is matched against a list of inflexions, using the same procedure (Fig. 2.d).

The problem is to find as quickly as possible the longest similar substrings at each point (i, j) of the analysis. We have no room here to go into technical details, but this is possible using rather sophisticated methods which consist of pre-computing tables from the similarly-pronounced relation between graphemes, and storing the dictionary in a coded form where each character is replaced by a code which stands for the longest substring which begins with this character and can be involved in some similarly-pronounced relation (Véronis, 1988b).

The restriction stipulated by Morgan (1970), and Durham *et al.* (1983) is that the unknown word must contain no more than *one* typographical mistake, since this will cover the large majority of cases : two typographical errors rarely occur in the same word (Pollock and Zamora, 1983). We soften this restriction by allowing one typographical error in the root, and another at the ending of the word, in the inflexion, while within a word we accept *an unlimited number of phonographic errors.* Words as incorrectly spelt as *ippeauttainnuz, hipptainuz, hyothénnuse* (for *hypoténuse*) are perfectly recognized. This algorithm is quite fast enough for natural-language interfaces using dictionaries stored in R.A.M., since the access time to the correct entry in a 300-word French dictionary generating 700 inflected forms is about 25 ms with a Pascal program on a Macintosh II computer. The time taken hardly depends at all on the length of the word or on the number of phonographic errors it contains. Better results could be obtained by a more sophisticated organization of the dictionary (in tree form, for example).

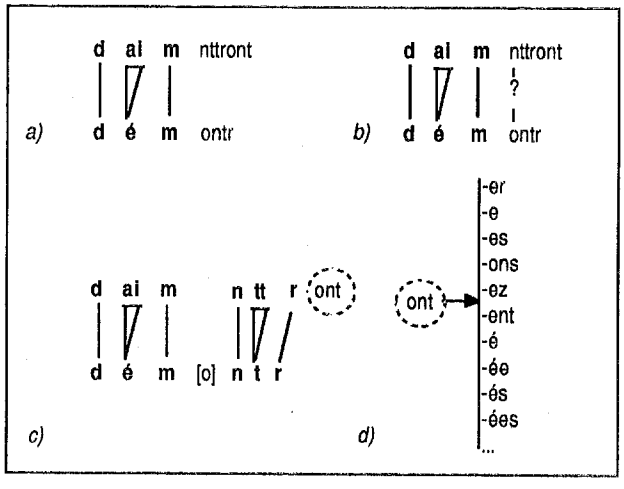

Figure 2 : phonographic correction of root and inflexion

III. Agreement correction

Once the right root and inflexion have been found in the dictionary during the lexical analysis, the morphological information (gender, number, etc.) associated with the word are passed on to the parser, which deals with any wrong agreements. In such a case, the parser builds various interpretations : *le triangles* ($the_{sg.}$ $triangles_{pl.}$) can be corrected into *les triangles* (plural) or into *le triangle* (singular). The problem is how to classify these interpretations depending on their plausibility. The few methods proposed so far (as in Richard and Lapalme, 1986) are not satisfactory. There are in fact two classical approaches. The first consists of favouring the interpretation which minimizes the total number of errors. For example, correcting *le triangles rectangles* ($the_{sg.}$ $right_{pl.}$ $triangles_{pl.}$) into *le triangle rectangle* (singular) implies two errors, whereas the correction into *les triangles rectangles* (plural) implies a single error. The second approach consists of always favouring the morphological features of fixed syntactic categories. For example, Richard and Lapalme (1986) propose favouring the determiner in French over the noun. This leads, in the previous example, to a correction into *le triangle rectangle* (singular). The two approaches are in many cases, as here, contradictory. One can use a combination of the two methods, for example by applying the second when the first fails (same number of errors upon each hypothesis), but this will not solve all problems. In fact, we needed to carefully investigate the agreement phenomena, in order to establish a weighting of errors.

Our first finding concerned the non-symmetry of errors. People very often forget unpronounced morphological markers but very rarely add them with no reason. Adding a marker costs more than removing it. Therefore, the group *triangles rectangle* ($right_{sg.}$ $triangles_{pl.}$) should be preferably corrected into *triangles rectangles* (plural).

One should also note the very important role of pronunciation. For example, it is very unlikely that a user might write *équilatéraux* ($equilateral_{pl.}$) for *équilatéral* ($equilateral_{sg.}$), since the two forms do not have the same pronunciation. Consequently, *triangle équilatéraux* ($equilateral_{pl.}$ $triangle_{sg.}$) should be preferably corrected into *triangles équilatéraux* (plural). In addition, one can assume that native speakers of French are unlikely to produce errors involving the *knowledge* of morphological features of words such as gender, number, person. Everybody knows that *chien* (dog) is masculine and *chienne* (female dog) is feminine. The difficulty is due to the *transcription* of agreement markers in an orthographical system. Therefore, errors such as *chienne dressé* ($trained_{masc.}$ $dog_{fem.}$) should be corrected into *chienne dressée* (feminine) and not into *chien dressé* (masculine). The situation would be different with non-native speakers of French, for example in a C.A.I. system for learning French, where gender errors would be very frequent. In this case, the weighting of errors would have to be different.

We postulate three classes of errors with increasing costs.

I. The least costly type of error consists of *deleting* a marker involving no change in the pronunciation (e.g. French *triangles* → *triangle*).

II. The second class consists of *adding* a marker which entails no pronunciation change (e.g. French *triangle* → *triangles*).

III. The third and most costly class consists of errors altering the pronunciation (e.g. *le* → *la*).

Some intermediate cases are distributed among these three classes. For example, errors involving a final so-called 'mute' *e* (which indicates the feminine, and has an unstable pronunciation) will belong to class II in the case of a deletion (e.g., *petite* → *petit* =small), and class III in the case of an addition (e.g., *petit* → *petite*).

The main point is that we cannot simply attibute an increasing weight to each class, and add the weights when combining phrases. It should be noted that an arbitrary number of errors in a given class remains less costly than a single error in the next class. For example,

les triangle rectangle et isocèle
($the_{pl.}$ $right_{sg.}$ and $isoceles_{sg.}$ $triangle_{sg.}$)

should be corrected into

les triangles rectangles et isocèles (plural)

with three class I errors, whereas the correction into

le triangle rectangle et isocèle (singular)

would involve a single error, but of class II.

This can be modelled by *ordinal numbers* : 0, 1, 2, 3..., ω, $\omega+1$..., ω^2, etc. (let us remember that $\omega^i.k < \omega^{i+1}$, $\forall$ k). Class I has costs of the form *k,* class II of the form $\omega.k$, and class III of the form $\omega^2.k$. In practice, ordinals can be coded by integers, by choosing a sufficiently large integer B (for example 10), and mapping $\omega^n.k'_n + ... + \omega.k'_1 + k'_0$ to $k'_nB^n + ... + k'_1B + k'_0$. For example, $\omega^2.2 + \omega.3 + 1$ will be coded by 231. This coding is adopted in the Figures.

The parser conducts the various possible morphological analyses in parallel, in order to avoid the costly backtrackings needed to repeat the analysis as soon as an error occurs, and also to avoid the need for any special error recovery procedure. This is achieved by associating a vector of the costs upon each possible morphological hypothesis with each node of the syntactic tree. The lexicon provides these values for each word

(figure 3). For example, the word *petits* ($small_{pl.}$) will be associated with the vector [ω,ω.2,0,1], which means that it can be a mistake for :

- *petit* (masc. sing.) with a cost ω (adding *s*)
- *petite* (fem. sing.) with a cost ω.2 (deleting mute *e* + adding *s*)
- *petits* (masc. plur.) with a cost 0 (no error)
- *petites* (fem. plur.) with a cost ω (deleting mute *e*).

In addition, each word is associated with a *domain,* which consists of the only possible corrections, since many words have restricted morphological features. This is the case with most nouns : *homme* (man) can be only masculine, *femme* (woman) only feminine, *gens* (people) only masculine plural, but also some adjectives : *enceinte* (pregnant) can be only feminine. We represent the domains by hatching the forbidden part, which is coded by a special value in the vector.

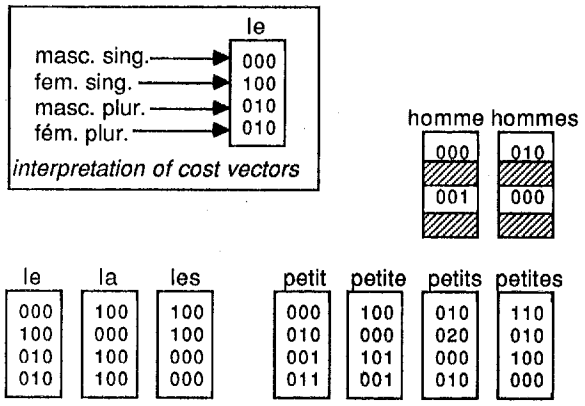

Figure 3 : cost vectors for words

When phrases are combined during the parsing, domains are intersected and costs are added separately in each vector column in the following way :

$$\begin{array}{lcccc} \alpha & = & \omega^2.k_2 & + \omega.k_1 & + k_0 \\ \beta & = & \omega^2.k'_2 & + \omega.k'_1 & + k'_0 \\ \hline \alpha \oplus \beta & = & \omega^2.(k_2+k'_2) & + \omega.(k_1+k'_1) & + (k_0+k'_0) \end{array}$$

Under the above-mentioned assumption for coding ordinals, the addition ⊕ can be reduced, in practice, to the ordinary addition of integers in base B. Therefore, the parallel computation of the various morphological hypotheses is not much more expensive than the usual exact, non-parallel, computation.

The same process can be applied to the other morphological features, persons, tenses and moods. In the final stage of parsing, the least costly hypothesis is chosen (Fig. 4, 5). If semantic constraints prove this interpretation to be impossible, the next hypothesis is chosen, and so on. This part is implemented in Prolog and calls on the Pascal module described in section II.

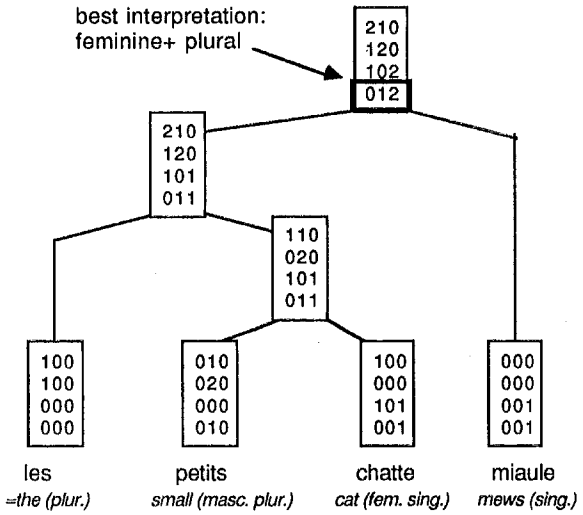

Figure 4 : agreement correction on a (very!) erroneous sentence.

IV. Conclusion

An efficient means of handling quite complex combinations of typographical, phonographic and agreement errors, which are frequent with C.A.I. users, is described : a sentence as erroneous as *les cotté adgassan à l'ippeauttainuz son perpndiqulère* (!) will be perfectly recognized and translated into *les côtés adjacents à l'hypoténuse sont perpendiculaires* (the legs adjacent to the hypotenuse are perpendicular). This feature can make interaction with systems more pleasant for non-specialists.

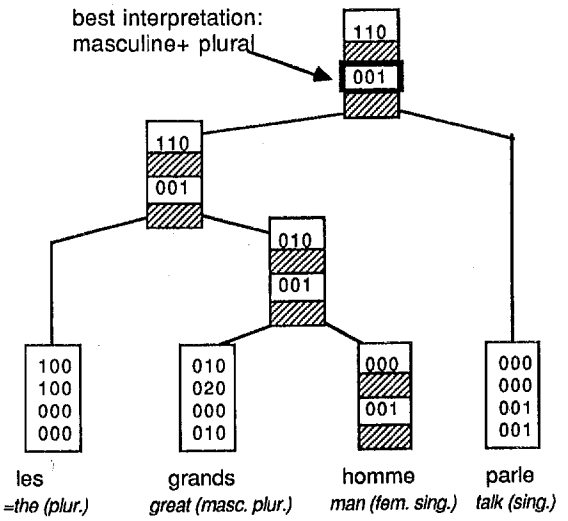

Figure 5 : Intersection of domains

REFERENCES

CHOMSKY, N. (1965). *Aspects of theTheory of Syntax.* Cambridge, Mass. : The MIT Press.

CHOURAQUI, E., INGHILTERRA, C., VÉRONIS, J. (1988). ARCHIMEDE : un système expert d'enseignement de la géométrie. *8th International Workshop Expert Systems and their Applications.* Avignon, France.

DAMERAU, D. N. (1964). A technique for computer detection and correction of spelling errors, *Comm. A.C.M.,* **7**, 3, 171-176.

DURHAM, I, LAMB, D. A., SAXE, J. B. (1983). Spelling correction in user interfaces, *Comm. A.C.M.,* **26**, 10, 764-773.

LOWRANCE, R., WAGNER, R. A. (1975). An extension to the string-to-string correction problem, *J. A.C.M.,* **22**, 2, 177-183.

MORGAN, H. L., (1970). Spelling correction in system programs, *Comm. A.C.M.,* **13**, 2, 90-94.

PETERSON, J. L. (1980). Computer programs for detecting and correcting spelling errors, *Comm. A.C.M.,* **23**, 12, 676-687.

POLLOCK, J. J. (1982) Spelling error detection and correction by a computer ; some notes and a bibliography, *J. Doc.,* **38**, 4, 282-291.

POLLOCK, J., J., ZAMORA, A. (1983). Collection and characterization of spelling errors in scientific and scholarly texts, *J. Am. Soc. Inf. Sc.,* **34**, 1, 51-58.

RICHARD, D., LAPALME, G. (1986). Un système de correction automatique des accords des participes passés. *Technique et Science Informatique,* **5**, 4, 307-320.

VERONIS, J. (1986). Etude quantitative sur le système graphique et phonographique du français. *European Bulletin of Cognitive Psychology,* **6**, 5, 501-531.

VERONIS, J. (1987 b). Discourse consistency verification and many-sorted logic. *Proceedings of the 10th International Joint Conference on Artificial Intelligence.* Milan, 633-635.

VERONIS, J. (1987 c). Vérification de cohérence dans le dialogue homme-machine en langage naturel. *Actes du Colloque Reconnaissance des Formes et Intelligence Artificielle, A.F.C.E.T.,* Antibes, 143-158.

VERONIS, J. (1988 a). Computerized correction of phonographic errors. *Computers and the Humanities,* **22**,1, 43-56.

VERONIS, J. (1988 b). Correction of phonographic errors in natural language interfaces. *11th International Conference on Research and Development in Information Retrieval.* Grenoble, France.

VERONIS, J. (1988 c). L'erreur dans le dialogue en langage naturel avec des systèmes experts. *8th International Workshop Expert Systems and their Applications.* Avignon, France.

VERONIS, J. (1988 d). Sound-to-spelling transcription : a computer simulation. *European Bulletin of Cognitive Psychology,* **8**, 3, [June 1988 : in press].

WAGNER, C. K., FISCHER, M. J. (1974). The string-to-string correction problem, *J. A.C.M.,* **21**, 1, 168-173.