

Classification Errors: a Useful Abstraction for Interpreting and Correcting Users' Mistakes

Authors: Jorge Marques Pelizzoni (contact author)
Maria das Graças Volpe Nunes

Address: NILC/ICMC – the University of São Paulo
Caixa Postal 668
São Carlos, Brasil, 13560-970

Phone No: 55-16-273-9628

E-Mail: {jorgemp, mdgvnune}@icmc.sc.usp.br

Abstract: A significant amount of the mistakes people make in everyday life can be regarded as ultimately resulting from classification errors. In this article, we approach the reversion of users' classification errors by (i) characterizing and contextualizing the problem, (ii) sketching a theoretical framework modeling classification as an error-prone operation and (iii) applying it to the spelling correction domain for Portuguese in a quick case study and thus covering a host of apparently heterogeneous morphology-motivated misspellings.

Keywords: Classification, Morphology, Portuguese, Spelling Error Correction.

Topic Areas: Natural Language Processing, Computational Morphology, Knowledge Representation, Grammar.

Conference Section: PAPER TRACK.

Under consideration for other conferences?

Yes (XVI Brazilian Symposium on Artificial Intelligence - SBIA'02).

Classification Errors: a Useful Abstraction for Interpreting and Correcting Users' Mistakes

Jorge Marques Pelizzoni and Maria das Graças Volpe Nunes

NILC/ICMC – the University of São Paulo
Caixa Postal 668
São Carlos, Brasil, 13560-970
{jorgemp, mdgvnune}@icmc.sc.usp.br

Abstract. A significant amount of the mistakes people make in everyday life can be regarded as ultimately resulting from classification errors. In this article, we approach the reversion of users' classification errors by (i) characterizing and contextualizing the problem, (ii) sketching a theoretical framework modeling classification as an error-prone operation and (iii) applying it to the spelling correction domain for Portuguese in a quick case study and thus covering a host of apparently heterogeneous morphology-motivated misspellings.

1 Introduction

A desirable though usually non-trivial object in computer science is the correction¹ of mistakes on the user's part when building some kind of artifact, i.e., the automatic generation of alternatives to an identified malformed item that keep the user's original intention with maximum probability. Examples of such systems are spell-checkers (Aurélio, 96; Microsoft, 97), grammar checkers (Lexikon, 97; DTS, 98; Itaotec, 99) e some compilers.

Correction systems usually prioritize the most often error types to the point of not tackling less frequent ones, except as a side effect. This seemingly sensible project directive, however, may well reveal itself fallacious to most interactive systems, to which the user's

(self-)correction facilities are also available. In such case, given a certain error, the weight (or, according to our terminology, the *utility*) of an accurate response by the system is proportional to the user's difficulty in correcting themselves. Unfortunately, in some applications, the most challenging error types are not noticeably often, resulting in the low utility of many correction systems, which end up providing actual help only when the user does not need any.

¹ “Error correction” refers here to an operation distinct to but preceded by “error identification”. It is worth mentioning that the latter is historically much simpler and more successful a task than the first (vide C compilers and spell-checkers). One of the reasons for this is the fact that identifying an error does not necessarily entail determining its causes or type.

This paper is a by-product of work on utility-driven spell-checking for Brazilian Portuguese, with a view to respond to a gap in an area that, according to Pelizzoni (02), has systematically neglected utility². In brief, our approach to maximizing this parameter was trying to identify, understand/model and so attack the different causes of spelling errors. As a result, several frameworks have been and are being developed, one of which we present here. In time, all examples of misspellings and their respective corrections are given in Brazilian Portuguese.

A host of misspellings obviously result from a mistaken operation in morphological processing, such as **cidadões* (*cidadãos* \cong “citizens”), **celebríssimo* (*celebérrimo* \cong “most famous”), **reaveu* (*reouve* \cong “[he] recovered [smth]”), **transporam* (*transpuseram* \cong “[they] transposed”), **di* (*dei* \cong “[I] gave”), **constrangiu* (*constrangeu* \cong “[he] embarrassed”), **diminói* (*diminui* \cong “[it] decreases”) and **vareia* (*varia* \cong “[it] varies”), as for inflection, and **planejação* (*planejamento* \cong “[the] planning”), **incortês* (*descortês* \cong “discourteous”) e **pré-câmara* (*antecâmara* \cong “antechamber”), as for derivation. Although all these examples involve errors in a range of morphological operations (either inflection or derivation; either verbal or nominal inflection; derivation either by suffixation or prefixation; etc.), an interesting result is that all of them could be suitably covered, in a uniform and elegant manner, when regarded as containing *not* morphological errors (in fact, all the examples above make “morphological” sense), *but* mistaken classification operations, or rather, *classification errors*.

This abstraction leap seems most relevant and valuable to us as we move from purely linguistic issues to others of interest to many domains. The aim of this paper is exactly to give an account of this experience, first sketching a theoretical model of classification as *an error-prone operation amenable to correction* and next demonstrating how this model is instantiated in our system, as a case study. This article is aimed at all those tackling the correction but, also and especially, the *interpretation* of users’ mistakes and searching for deeper patterns in a universe of errors.

2 A Model of Classification

2.1 (Classification) Errors: Optimism, Depth, Intention, Reversal, Potential Confusion, Cats & Microwaves

Any malformed item amenable to correction is much more of a hit than a miss: there is hope if and only if the user has been much more right than wrong in the production of such an item. It is starting from this *optimism* that any correction is made possible. The distinction should be clear hereafter between “malformed item” and “error”: the first is the observable result of a process in which the latter occurs as a disturbing

² Several spell-checkers have had their design based on Damerau’s (64) result that 80% of all misspellings in English typed texts contain one single instance of a *simple error* (insertion, deletion or substitution of one character or swapping of two characters).

factor of, hopefully, restricted scope. This vision of “error” as “a mistaken operation in a process” as opposed to the usual “a defect in a product” or “a defective product” is rather lucid and opportune, providing for a suitable model of correction.

One first implication of this change of perspective is that there is no such thing as *surface errors*: rather, every error is at a certain *depth*, i.e., it is never explicit on the malformed item so that its identification dispenses with some sort of inference, supposition or analysis regarding a production process. The simplest spell-checkers, for instance, assume that all errors occur in typing, the most superficial level of typed word production. In fact, these systems regard the process as starting from the correct spelling of the intended word (!), hence the essence of their low utility.

A great deal of the mistakes people make in everyday life can be regarded as ultimately stemming from classification errors. The word “ultimately” is especially significant here and accurately expresses the usual depth of such errors. Naturally, the disaster is not the idea proper that a cat is an instance of class *Microwaveable*³, but what is made of it *from* this fancy idea. That is, in such a situation, a correction system would sense the presence of an error by the user’s dissatisfaction to see what became of their feline friend after microwave treatment. What was the actual error – the *point* in the process that triggered the “malformed” cat – and how could it be *reverted*?

The reversal of this error, of course, could never consist of the cat’s resurrection. Neither of informing the user that cats are murdered that way, which lesson he would have just learned. One accurate alternative would be the emission, for example, of the following enlightening suggestion: “Next time try a hairdryer... or a towel!” However, how can a system reach that conclusion without the user being able to express their *original intention*? For this condition, here apparently absurd, is all too true in the analogous situations that a computer user goes through. In most cases, the user being able to inform the computer their original intention implies that they are also able to correct themselves alone⁴.

The processing to produce the desired output is not trivial at all. One possible (and oversimplified) thread of reasoning is shown in Figure 1.

³ Microwaveable, *adj.* that emerges from a microwave oven in better shape.

⁴ Reciprocally, the computer understanding such a communication would imply such (artificial) intelligence as to make the present discussion and project obsolete. At the most, the system may present (few, good and clear) alternatives of original intentions from which to choose.


CONCLUSIONS/QUESTIONS	RELEVANT FACTS
	The user loved the cat.
SO the user did not mean to kill her. What did he mean then? Heat the cat AND keep her alive. What for?	
	Heat dries water AND the cat was wet AND humans like it dry AND time is money.
To dry her fast. What can help dry the cat fast without killing her? A microwave hairdryer!	 High potential confusion!

Fig. 1. Reversal⁵ of the cat’s drying/sacrificing process

In order to suggest that the user should try a hairdryer, the system would undoubtedly have to be optimistic and infer the user’s original intention from the (assumed) right steps in the process. Moreover, it would probably have to hypothesize that user asked themselves the question “What can help dry the cat...” and made a mistake upon answering it. Stating the problem in terms of classification, the user failed upon classifying a microwave oven as an instance of *SafeDryer*, the class of all equipment that can be used to dry living creatures (and keep them that way!).

It is worth noticing, finally, that the supposed classification error is not fortuitous. Microwave ovens and *SafeDryers* have lots in common, enough, in fact, to mislead the user. One could say that the accident at issue has some intrinsic potentiality or that there is *high potential confusion* between those two classes.

In summary and principle, the procedure of error reversal can be conceived of as involving two steps:

- the reversal⁵ of the process that yielded the malformed item. This step may well result in various hypothetical reconstructions as most of the process, if not all of it, usually happens exclusively in the user’s mind. Good reconstructions will always be optimistic – containing just a few *error points*– and consistent with all the known circumstances;
- the replay of the reconstructions so obtained, now revised as for the results of the (assumed) mistaken operations.

What is shown in Figure 1 is simply one hypothetical reconstruction of the cat’s sacrificing process, which is considered excellent because it is based on the assumption of a single error, in a classification operation with high potential confusion. The more confusion involved the more plausible a classification error. Next we detail some of the concepts just introduced.

⁵ “Reversal” should be interpreted here as “backward reconstruction”, rather in the “police” sense of the term.

2.2 Classification: a Potentially Confusing Operation

A first trap to avoid when attempting to model classification is reduce it to the mere evaluation of the truth-value of a predicate *instance/2* as follows:

$$\text{instance}(O, C) \leftrightarrow \text{object } O \text{ is an instance of class } C, \quad (1)$$

or yet define it as something like:

$$\text{classes}(O) = \{c \in \mathbf{U} \mid \text{instance}(O, c)\}. \quad (2)$$

The first version naïvely deprives classification of its character of “selection from a set of possible classes”. Function *classes(x)*, in turn, respects this character, but is defective in that it does not restrict the universe of possible classes, i.e., it does not contextualize the choice. Let us see a more suitable version: given an object *O* and any set of classes *Context*, the (operation of) **classification** of *O* in *Context*, denoted by *classes(O, Context)*, is the set

$$\text{classes}(O, \text{Context}) = \{c \in \text{Context} \mid \text{instance}(O, c)\}. \quad (3)$$

That is, the set of all classes in *Context* that have *O* as an instance. Notice that the classification may be empty and that the following equivalence holds:

$$\text{instance}(O, C) \equiv [\text{classes}(O, \{C\}) = \{C\}]. \quad (4)$$

Any shadow of preciousness fades away when one intuitively compares the following classifications as to potential confusion:

- *classes*(scorpion, {number, letter, color}),
- *classes*(scorpion, {arachnid, mammal}) and
- *classes*(scorpion, {insect, arachnid, crustacean, arthropod});

Furthermore, the following mistakes:

- **classes*(scorpion, {car, bicycle}) = {bicycle} and
 - **classes*(scorpion, {insect, arachnid, crustacean, arthropod}) = {insect}
- have flagrantly disparate plausibilities. In the “microwaved cat” example (previous section), the error could be expressed as follows:
- **classes*(microwave, {SafeDryer}) = {SafeDryer}.

Established the role of context in a classification operation, we move on to consider formal prototypes to (i) the *plausibility of a classification error* and (ii) *confusion*. It may appear, at first, that plausibility depends exclusively on the confusion inside the set of classes from which to choose in a classification operation. This idea is ruled out when confronting such pairs as:

- **classes*(scorpion, {insect, arachnid, crustacean, bicycle}) = {bicycle} and
- **classes*(scorpion, {insect, arachnid, crustacean, bicycle}) = {crustacean}.

The great contrast between the examples above is due to the fact that scorpions are in many respects *similar* to crustaceans, whereas they can hardly be *compared* to bicycles. Naturally comes into play the very core module of classification, namely comparison between objects, involving not only factoring common features but also identifying differences. In our modeling, the *front-end* of this module is a function

confusion: $P(U_{classes})^6 \rightarrow [0, 1]$, such that *confusion*(x) is the *degree of confusion/similarity/uniformity between the classes in x* .

Counting on a good *confusion*, one factor with an influence on the plausibility of any classification error $\star classes(O, Context) = Classes$ is

$$confusion(Classes \cup \{<O>^7\}) . \quad (5)$$

This factor still needs contextualizing. To this end, we opted to contrast it with the sum of the factors for all possible results (one alternative is to consider the maximum factor only), except for the empty result, which we treat separately. This is implemented in the following definition:

$$plausibility(O, Cntxt, Rslt) = \frac{factor(O, Cntxt, Rslt)}{\sum_{any \in P(Cntxt)} factor(O, Cntxt, any)} \quad (6)$$

where

$$factor(O, Cntxt, Rslt) = \begin{cases} 1 - \underset{Cs \in P(Cntxt)}{Max} \{confusion(Cs \cup \{<O>\})\}, & Rslt = \emptyset \\ confusion(Cs \cup \{<O>\}), & \forall Rslt \neq \emptyset. \end{cases}$$

All that remains to be done now is define a reasonable *confusion*. As any such function must analyze a set of classes, it is in order to decide how to “implement” the class concept. Only for illustrative purposes, let classes be predicate sets, the following definitions holding:

$$instance(O, C) \leftrightarrow \forall p [p \notin C \vee p(O)] . \quad (7)$$

That is, an object O is considered an instance of a class C iff all predicates in C hold for O . Furthermore, let A and B be classes:

$$A \dashv B = \{p \in A \mid \sim \exists q (q \in B \wedge q \equiv p)\} . \quad (8)$$

$$A .\Lambda. B = \{p \in A \mid \exists q (q \in B \wedge q \equiv p)\} . \quad (9)$$

$$A .U. B = (A \dashv B) \cup (B \dashv A) \cup (A .\Lambda. B) . \quad (10)$$

The three binary class operators defined above are respectively analogous to the usual three binary set operators – namely difference, intersection and union – suitably adapted to handle predicate equivalence. Follows a first prototype of *confusion*:

$$confusão(\{c_i\}) = \frac{\#(\underset{i=1}{\overset{n}{\Lambda}} c_i)}{\#(\underset{i=1}{\overset{n}{U}} c_i)} \quad (11)$$

⁶ $P(X)$ is the set of all subsets of X . U_x , in turn, is the universe of all x and, for $x = classes$, denotes the set of all existing classes.

⁷ The operation $<X>$ denotes the conversion of entity X into a class, i.e., $<X>$ is a computed class that has X as an instance and is the most specific possible. This operation may not be trivial depending on how the concept of class is “implemented”.

The version above, naïve though it may be, is essentially perfect, growing with the number of features that are common to all classes, but with a sense of proportion. Nonetheless, it is not realistic when it considers all features equally relevant, i.e., with the same weight. It is a fact that certain features of a class are felt to be more “characteristic” than others. For example, we believe that “produces milk” is generally considered rather more characteristic of mammals than “has hot blood”, even though, strictly speaking, both features are required of any candidates to mammals. We therefore assume a function $e: (U_{\text{predicados}} \times P(U_{\text{classes}})) \rightarrow [0, 1]$ that calculates the *membership* of features (predicates) to contexts (sets of classes). One way to define this function for non-singleton contexts is:

$$e(p, \{c_1, c_2, \dots, c_n\}) = \frac{\sum_{i=1}^n e(p, \{c_i\})}{n} \quad (12)$$

which computes but a simple mean value and reduces the problem to the calculation of membership to singleton contexts. The latter, nevertheless, is non-trivial, varying with application domains and ultimately related with users’ idiosyncrasies.

Now we can release a reasonable version of *confusion*, first just defining function $eTotal$ as a mere notational aid that adds up the memberships of a set of predicates (class) to a context, like this:

$$eTotal(Classe, Contexto) = \sum_{p \in Classe} e(p, Contexto) \quad (13)$$

$$confusão(C_s) = \frac{eTotal(\Lambda, c, C_s)}{eTotal(\mathbb{U}, c, C_s)} \quad (14)$$

3 Morphology and Classification: a Case Study

The idea of classification is in no way strange to morphology. It suffices to mention that concepts like *paradigm* and *model* have currency in the literature on morphology (Monteiro, 86) and that words are grouped (i.e., classified) according to their adherence to this or that paradigm or model. Among the abundant Portuguese morphological classes are “adjectives whose superlative is made adding *-íssimo/-érrimo*”, “verbs that are inflected like *cantar/vender/partir/pôr/passear/odiar/construir/etc.*”, “verbal themes that make abstract deverbatives in *-ção/-mento*” and so on. Naturally, classification errors are expected inside any of these three class sets (contexts). Nonwords like \star *conjugamento* (*conjugação* \cong “inflection”), \star *vareia* (*varia* \cong “[it] varies”) e \star *diminói* (*diminui* \cong “[it] decreases”) can be accounted for by the following classification errors:

- \star classes(conjuga, {<-ção>, <-mento>}) = {<-mento>};
- \star classes(variar, {<cantar>, <odiar>}) = {<odiar>};
- \star classes(diminuir, {<partir>, <construir>}) = {<construir>}.

In order to generate hypotheses of word formation containing this kind of errors and next perform any applicable corrections, we opted to represent the necessary knowledge by means of a unification-based (Shieber, 86) word grammar (Agirre et al., 92; Sengupta & Chaudhuri, 96), according to a model inspired by *LFGs*⁸, though rather simplified and also extended so as to support paradigms. The referred simplification consists of allowing one single level of unification, i.e., all variables are global. Figure 2 presents a code sample in our formalism to give an idea of how these features are realized.

```

verbo --> tema_verbal, flexao.
tema_verbal --> radical_verbal, vt. /* vt: vogal temática */
flexao --> dmt, dnp. /* desinências modo-temporal e número-pessoal */

paradigm tempos_primitivos.
  dmt --> {Ø}, [tm = pret_perf/ind, np = not(3/plural)].
  ... end.
paradigm conjI extends tempos_primitivos.
  vt --> {a}.
  dnp --> {i-assilabico}, [tm = pret_perf/ind, np = 1/sing].
  ... end.
paradigm conjIIouIII extends tempos_primitivos.
  dnp --> {i-silabico-tonico}, [tm = pret_perf/ind, np = 1/sing].
  ... end.
paradigm conjII extends conjIIouIII. vt --> {e}. ... end.
paradigm conjIII extends conjIIouIII. vt --> {i}. ... end.

```

legenda: { ... }	= símbolo terminal	np = pessoa/número
[...]	= casamento de variáveis	tm = tempo/modo

Fig. 2. A sample of the grammatical formalism

The convenience of a unification-based formalism, in this application, is the natural expression of agreement constraints, which also occur at the level of morphology. By means of variable unification, the various grammatical features (gender, number, time, mood, person, etc.) are retrieved, providing for the correction of inflection errors. Similarly, grammatical and (to a lesser and tentative extent) some semantic features associated with derivational morphemes (lexical prefixes and suffixes) are treated with a view to derivation errors.

As shown in Figure 2, generalization/specialization can also be naturally expressed, the keyword **extends** introducing superclass lists. Classes here are to be regarded simply as hierarchical production rule blocks. Accordingly, there is an important restriction: subclasses must inherit superclass behavior integrally, i.e., *overriding* is banned, only extension is allowed. The semantics of the class concept in our formalism is simple but effective as follows: if, on parsing/generating a word, some rule in class *C* is applied, then every rule in the hierarchy⁹ of *C*, except for its direct or not superclasses and subclasses, is considered inapplicable. Thus, non-terminals on the right-hand side of the rules in a given class refer to entities defined (i) in its “ancestors” and “descendants”, (ii) globally or (iii) in classes of other hierarchies.

⁸ Lexical-Functional Grammars.

⁹ The *hierarchy* of a class *C* is the set of all classes that have some “kinship” to *C*, or rather, all those that have some superclass in common with *C*.

3.1 From $\star di$ to \check{dei}

So as to demonstrate how the formalism above, together with the ideas presented in Section 2, can be used in the correction of word formation mistakes, we will trace one single though sufficient example, since processing in the remaining cases is analogous (and usually simpler) and publication space, restricted. The malformed word in question is $\star di$ ($dei \cong$ “[I] gave”), not often in writing, but most revealing.

The reversal process is triggered as soon as the system concludes that $\star di$ does not belong to the lexicon¹⁰. As a result, a series of reconstruction hypotheses start to be considered, also supposing errors on levels other than morphology that are irrelevant to the present discussion. Resorting to knowledge as presented in a simplified form in Figure 2 and a suitable *bottom-up parser*, two partial hypotheses of word formation are generated for $\star di$, schematically shown as follows:

- H_{conjII} : $d_{verb-stem} + \langle \star conjII \rangle + e_{vt} + \emptyset_{dmt} + i\text{-silabic-stressed}_{dnp}$;
- $H_{conjIII}$: $d_{verb-stem} + \langle \star conjIII \rangle + i_{vt} + \emptyset_{dmt} + i\text{-silabic-stressed}_{dnp}$.

The *parser* stops precisely at these points because it is instructed to try correction at every *class decision point*¹¹, marked with $\langle \star \rangle$ above. Both hypotheses assume that class decision is mistaken; but not the user’s *original intention*, which has been partially retrieved into variables np e tm . Again in both cases, these variables inform that the user must have intended a 3rdpsing past simple indicative form of a supposed verb of stem “d”.

At this point, we meet a singular situation, remarkably distinct from those in Section 2.2, for nothing is known about the supposed verb – the *object* that is being classified. That is, it is presently impossible to estimate the plausibility of these classification errors. Our framework, however, is not nullified: the correction system simply assumes some high plausibility to be verified *a posteriori*. This assumption is based on a valuable result: whatever the correct verb that the user should have used, *it must necessarily have heavy-weighted features in common with the supposed verb that has been inflected in the wrong classes*.

These features are a fixed and restricted set of the most characteristic/most often used verbal inflections (remember that we are dealing with contexts/sets of verb classes). We could have feature membership e as defined in Table 1, where Cx is the classification context, which is trivially obtained as the hierarchies of the classes involved (either *conjII* or *conjIII*). In this specific example, both hypotheses happen to have one same context, namely *temposPrimitivos* and all of its subclasses.

¹⁰ We have a highly compacted (less than 1,5Mb), comprehensive (over 1,500 thousand entries, each grammatically tagged) lexicon that is accessed with extreme efficiency (constant time). In addition, there are relationships between entries that allow inflection and lemmatization with the same efficiency.

¹¹ A *class decision point* is a moment in word *generation* at which the generator/user has no choice but to decide for a specific class in order to proceed. In the example being traced and supposing left-to-right *generation* (natural of suffixation/inflection), the only such point occurs just after parsing non-terminal *vt*.

Table 1. Definition of $e(x, Cx)$ and related inflections in each classification error hypothesis

x (variable set-up)	$e(x, Cx)$	Inflecting $d+<*\text{ConjII}>$	inflecting $d+<*\text{ConjIII}>$
[tm = inf_ impessoal, np = 0]	1	*der	*dir
[tm = pres/ind, np = 1/sing]	1	*do	*do
[tm = pret perf/ind, np = 1/sing]	1	*di	*di
[tm = pret perf/ind, np = 3/sing]	1	✓deu	*diu
[tm = participio, gn = masc/sing]	1	*dido	*dido
Outros	0	–	–

Let us first consider H_{conjII} . In order to test this hypothesis, the system verifies which relevant features of class *ConjII* is also present for some lexicalized verb. This involves, for each variable set-up X , (i) starting *generation* from where *parsing* stopped, requiring X to be in the final configuration, (ii) consulting the lexicon for the resulting *string* (third column in Table 1) and, in case of success, (iii) verifying if the grammatical features of this *word* correspond to those in X and in the non-terminals themselves of the word grammar (in the example, we cannot forget that we are dealing with **verb**[o]s).

In testing H_{conjII} , as shown in Table 1, this procedure yielded one single *link feature* – the inflection “deu” (\cong [he] gave) – between *der and some correct verb. The high membership of this feature is enough to validate the system’s early assumption of high error plausibility. Finally, the last correction step consists of asking the lexicon to inflect “deu” according to the user’s original intention, yielding “dei”, a good suggestion of correction.

What could perhaps be considered utterly absurd, namely inflecting “dar” as a verb of the 2nd Paradigm (strongly characterized by *vt* “e”), proves itself realistic. In fact, the error in *di, as well as most morphology-motivated mistakes, results from an act of intelligence: the analogy “[ele] vendeu is to [ele] deu as [eu] vendi is to [eu] di” is perfect and reveals that similarity between infinitives is just one among the numerous sources of confusion in verb inflection.

The test of $H_{conjIII}$ follows the same procedure as H_{conjII} , not yielding, however, lexicalized strings, which nullifies the early assumption of plausibility for $H_{conjIII}$ *a posteriori*. It is worth mentioning that the mistake has been corrected in spite of a complete derivation tree having never been built, which is one of the reasons why we opted for *bottom-up* analysis.

Further interesting examples that can be traced likewise are *constrangiu (*constrangeu* \cong “[he] embarrassed”) e *reaveu (*reouve* \cong “[he] recovered”), in which cases the link features are “constrangido” and “reaver”, respectively. Errors such as that in *vareia (*varia* \cong “[it] varies”) are also analogous, just requiring an extension of the grammar introducing new paradigms (that of “odiar”, in this specific case).

4 Conclusion

Reasonable as it may seem, the approach to morphology-motivated error correction presented here has not as yet been evaluated, as a working prototype of the correction system is not fully functional. Naturally, unforeseen side effects, especially resulting from the interaction with other system modules, might lead to revisions of the model, mainly as regards its procedural specification. Notwithstanding, we firmly believe that (i) the level of abstraction achieved over our original error types and (ii) current results in knowledge representation are accomplishments most unlikely to be nullified. Realism and simplicity are further strengths we identify in our representation of morphological paradigms, especially in the inflectional ones, in which the base form is not the only starting point for inflection.

References

- (Agirre et al., 92) Agirre, E., Alegria, I., Arregi, X., Artola, X., Díaz De Ilarraza, A., Maritxalar, M., Sarasola, K., Urkia, M. XUXEN: A Spelling Checker/Corrector for Basque Based on Two-Level Morphology. In *3^a Conf. of Applied NLP*, 1992, 119-125.
- (Almeida & Pinto, 95) Almeida, J. J., Pinto, U. Jspell — um módulo para análise léxica genérica de linguagem natural. In *Actas do Congresso da Associação Portuguesa de Linguística*, Évora, 1995.
- (Aurélio, 96) *Dicionário Aurélio Eletrônico*. Versão 2.0. Copyright © 1996 Editora Nova Fronteira.
- (Damerau, 64) Damerau, F. J. A Technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7, 3 (mar./1964), 171-176.
- (DTS, 98) *Revisor Gramatical DTS*. Versão 3.0. Copyright ©1998 DTS Software.
- (Itautec, 99) *Redação Língua Portuguesa*. Versão 7.1. Copyright ©1995-1999 Itautec-Philco.
- (Lucchesi & Kowaltowski, 93) Lucchesi, C. L., Kowaltowski, T. Applications of Finite Automata Representing Large Vocabularies. *Software — Practice and Experience*, 23, 1 (1993), 15-30.
- (Lexikon, 97) *Gramática Eletrônica*. Versão 1.0. Copyright ©1997 Lexikon Informática.
- (Lins et al., 99) Lins, R. D., Carnelo, H. ^a L., Moura, R. S. Um SOS para a Língua Portuguesa. In *Actas do IV Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR'99)*, 1999, 129-138.
- (Microsoft, 97) *Microsoft® Word 97* (Editor de textos que embute um corretor ortográfico). Copyright ©1983-1997 Microsoft Corporation.
- (Monteiro, 86) Monteiro, J. L. *Morfologia Portuguesa*. Editora da Universidade Federal do Ceará (EUFCE), 1986.
- (Pacheco, 96) Pacheco, H. C. F. *Uma Ferramenta de Auxílio à Redação*. Dissertação de Mestrado, Departamento de Ciência da Computação, Instituto de Ciências Exatas, UFMG, 1996.
- (Pelizzoni, 02) Pelizzoni, J.M. *Preâmbulo ao aconselhamento ortográfico para o português do Brasil — Uma releitura baseada em utilidade e conhecimento lingüístico*. MSc. Thesis. Instituto de Ciências Matemáticas de São Carlos, USP. Apr, 2002.
- (Sengupta & Chaudhuri, 96) Sengupta, P., Chaudhuri, B. Morphological Processing of Indian Languages for Lexical Interaction with Application to Spelling Error Correction. In *Sadhana-Academy Proceedings In Engineering Sciences*, 21, Part 3, Jun. 1996, 363-380.
- (Shieber, 86) Shieber, S. M. *An Introduction to Unification-based Approaches to Grammar*. CSLI Lecture Notes Series, Chicago: University of Chicago Press.