Automatic Spelling Correction in Galician[*]

M. Vilares[1], J. Otero[1], F.M. Barcala[2], and E. Domínguez[2]

[1] Department of Computer Science, University of Vigo
Campus As Lagoas s/n, 32004 Ourense, Spain
{vilares,jop}@uvigo.es

[2] Ramón Piñeiro Research Center for Humanities
Estrada Santiago-Noia, Km. 3, A Barcia, 15896 Santiago de Compostela, Spain
{fbarcala,edomin}@cirp.es

Abstract. We describe a proposal on spelling correction intended to be applied on Galician, a Romance language. Our aim is to put into evidence the flexibility of a novelty technique that provides a quality equivalent to global strategies, but with a significantly minor computational cost. To do it, we take advantage of the grammatical background present in the recognizer, which allows us to dynamically gather information to the right and to the left of the point at which the recognition halts in a word, as long as this information could be considered as relevant for the repair process. The experimental tests prove the validity of our approach in relation to previous ones, focusing on both performance and costs.

1 Introduction

Galician belongs to the group of Latin languages, with influence of peoples living here before the Roman colonization, as well as contributions from other languages subsequent to the breaking-up of this empire. Long time relegated to informal usage, it has managed to survive well into the 20^{th} century until it was once again granted the status of official language for Galicia, together with Spanish. Although there several dialects exist, it has been recently standardized and, as a consequence, there is a pressing need for tools in order to permit a correct linguistic treatment. A main point of interest is the development of efficient error repair tools, in particular for spelling correction purposes.

In this context, the state of the art focuses on global techniques based on the consideration of error thresholds to reduce the number of repair alternatives, a technique often dependent on the recognizer. So, Oflazer [5] introduces a *cut-off distance* that can be performed efficiently by maintaining a matrix [2] which help the system to determine when a partial repair will not yield any result by providing non-decreasing repair paths. In order to save this maintaining, Savary [6] embeds the distance in the repair algorithm, although this allows to partial corrections may be reached several times with different intermediate

[*] Research partially supported by the Spanish Government under projects TIC2000-0370-C02-01 and HP2002-0081, and the Autonomous Government of Galicia under projects PGIDIT03SIN30501PR and PGIDIT02SIN01E.

J. L. Vicedo et al. (Eds.): EsTAL 2004, LNAI 3230, pp. 45–57, 2004.

distances; which is not time-efficient for error threshold values bigger than one. Anyway, these pruning techniques are strongly conditioned by the estimation of the repair region and their effectiveness is relative in global approaches.

In contrast to global algorithms, that expend equal effort on all parts of the word, also on those containing no errors; we introduce regional repairs avoiding to examine the entire word. This is of importance since Galician is an inflectional language with a great variety of morphological processes, and a non-global strategy could drastically reduce the costs. In effect, work underway focusing on word processing, the descriptive model is a regular grammar (RG) and the operational one is a finite automaton (FA). At this point, repairs on RG's are explored breadth-wise; whilst the number of states in the associated finite automaton (FA) is massive. So, a complex morphology impacts both time and space bounds, that can even become exponential; which justifies our approach.

2 The Error Repair Model

Our aim is to parse a word $w_{1..n} = w_1 \ldots w_n$ according to a RG $\mathcal{G} = (N, \Sigma, P, S)$. We denote by w_0 (resp. w_{n+1}) the position in the string, $w_{1..n}$, previous to w_1 (resp. following w_n). We generate from $\mathcal{G}$ a *numbered minimal acyclic finite automaton* for the language $\mathcal{L}(\mathcal{G})$. In practice, we choose a device [4] generated by Galena [3]. A FA is a 5-tuple $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{Q}_f)$ where: $\mathcal{Q}$ is the set of states, Σ the set of input symbols, δ is a function of $\mathcal{Q} \times \Sigma$ into $2^{\mathcal{Q}}$ defining the transitions of the automaton, q_0 the initial state and $\mathcal{Q}_f$ the set of final states. We denote $\delta(q, a)$ by $q.a$, and we say that $\mathcal{A}$ is *deterministic* iff $\mid q.a \mid \leq 1,\ \forall q \in \mathcal{Q},\ a \in \Sigma$. The notation is transitive, $q.w_{1..n}$ denotes the state $(\overset{n}{\ldots}\ (q.w_1)\ \overset{n-1}{\ldots}).w_n$. As a consequence, w is *accepted* iff $q_0.w \in \mathcal{Q}_f$, that is, the *language accepted by* $\mathcal{A}$ is defined as $\mathcal{L}(\mathcal{A}) = \{w,$ such that $q_0.w \in \mathcal{Q}_f\}$. A FA is *acyclic* when the underlying graph it is. We define a *path in the* FA as a sequence of states $\{q_1, \ldots, q_n\}$, such that $\forall i \in \{1, \ldots, n-1\},\ \exists a_i \in \Sigma,\ q_i.a_i = q_{i+1}$. In order to reduce the memory requirements, we apply a minimization process [1]. Two FA's are *equivalent* iff they recognize the same language. Two states, p and q, are *equivalent* iff the FA with p as initial state, and the one that starts in q recognize the same language. An FA is *minimal* iff no pair in $\mathcal{Q}$ is equivalent.

2.1 The Dynamic Programming Frame

Although the standard recognition process is deterministic, the repair one could introduce non-determinism by exploring alternatives associated to possibly more than one recovery strategy. So, in order to get polynomial complexity, we avoid duplicating intermediate computations in the repair of $w_{1..n} \in \Sigma^+$, storing them in a table $\mathcal{I}$ of *items*, $\mathcal{I} = \{[q, i],\ q \in \mathcal{Q},\ i \in [1, n+1]\}$, where $[q, i]$ looks for the suffix $w_{i..n}$ to be analyzed from $q \in \mathcal{Q}$.

We describe our proposal using *parsing schemata* [7], a triple $\langle \mathcal{I}, \mathcal{H}, \mathcal{D} \rangle$, with $\mathcal{H} = \{[a, i],\ a = w_i\}$ an initial set of items called *hypothesis* that encodes the word to be recognized[1], and $\mathcal{D}$ a set of *deduction steps* that allow to derive items from previous ones. These are of the form $\{\eta_1, \ldots, \eta_k \vdash \xi \,/ \mathit{conds}\}$, meaning that if all antecedents η_i are present and the conditions *conds* are satisfied, then the consequent ξ is generated. In our case, $\mathcal{D} = \mathcal{D}^{\text{Init}} \cup \mathcal{D}^{\text{Shift}}$, where:

$$\mathcal{D}^{\text{Init}} = \{\, \vdash [q_0, 1] \,\} \qquad \mathcal{D}^{\text{Shift}} = \{\, [p, i] \vdash [q, i+1] \,/ \exists [a, i] \in \mathcal{H},\ q = p.a \,\}$$

The recognition associates a set of items S^w_p, called *itemset*, to each $p \in \mathcal{Q}$; and applies these deduction steps until no new application is possible. The word is recognized iff a *final item* $[q_f, n+1]$, $q_f \in \mathcal{Q}_f$ has been generated. We can assume, without lost of generality, that $\mathcal{Q}_f = \{q_f\}$, and that exists an only transition from (resp. to) q_0 (resp. q_f). To get this, we augment the FA with two states becoming the new initial and final states, and relied to the original ones through empty transitions, our only concession to the notion of minimal FA.

2.2 The Formalization

Let's assume that we deal with the first error in a word $w_{1..n} \in \Sigma^+$. We extend the item structure, $[p, i, e]$, where now e is the error counter accumulated in the recognition of w at position w_i in state p. We talk about the *point of error*, w_i, as the point at which the difference between what was intended and what actually appears in the word occurs, that is, $q_0.w_{1..i-1} = q$ and $q.w_i \notin \mathcal{Q}$. The next step is to locate the origin of the error, limiting the impact on the analyzed prefix to the context close to the point of error, in order to save the computational effort.

Since we work with acyclic FAs, we can introduce a simple order in $\mathcal{Q}$ by defining $p < q$ iff exists a path $\rho = \{p, \ldots, q\}$; and we say that q_s (resp. q_d) is a *source* (resp. *drain*) for ρ iff $\exists a \in \Sigma,\ q_s.a = p$ (resp. $q.a = q_d$). In this manner, the pair (q_s, q_d) defines a *region* $\mathcal{R}^{q_d}_{q_s}$ iff $\forall \rho,\ \text{source}(\rho) = q_s$, we have that $\text{drain}(\rho) = q_d$ and $\mid \{\forall \rho,\ \text{source}(\rho) = q_s\} \mid > 1$. So, we can talk about *paths*$(\mathcal{R}^{q_d}_{q_s})$ to refer the set $\{\rho / \text{source}(\rho) = q_s,\ \text{drain}(\rho) = q_d\}$ and, given $q \in \mathcal{Q}$, we say that $q \in \mathcal{R}^{q_d}_{q_s}$ iff $\exists \rho \in \text{paths}(\mathcal{R}^{q_d}_{q_s}),\ q \in \rho$. We also consider $\mathcal{A}$ as global region. So, any state, with the exception of q_0 and q_f, is included in a region. This provides a criterion to place around a state in the underlying graph a zone for which any change applied on it has no effect over its context. So, we say that $\mathcal{R}^{q_d}_{q_s}$ is the *minimal region in* $\mathcal{A}$ *containing* $p \in \mathcal{Q}$ iff it verifies that $q_s \geq p_s$ (resp. $q_d \leq p_d$), $\forall \mathcal{R}^{p_d}_{p_s} \ni p$, and we denote it as $\mathcal{M}(p)$.

We are now ready to characterize the point at which the recognizer detects that there is an error and calls the repair algorithm. We say that w_i is *point of detection* associated to a point of error w_j iff $\exists q_d > q_0.w_{1..j},\ \mathcal{M}(q_0.w_{1..j}) = \mathcal{R}^{q_d}_{q_0.w_{1..i}}$, that we denote by *detection*$(w_j) = w_i$. We then talk about $\mathcal{R}^{q_d}_{q_0.w_{1..i}}$ as the *region defining the point of detection* w_i. The error is located in the left recognition context, given by the closest source. However, we also need to locate it from an operational viewpoint, as an item in the process. We say that $[q, j] \in S^w_q$ is an *error item* iff $q_0.w_{j-1} = q$; and we say that $[p, i] \in S^w_p$ is a *detection item* associated to w_j iff $q_0.w_{i-1} = p$.

[1] A word $w_{1...n} \in \Sigma^+$, $n \geq 1$ is represented by $\{[w_1, 1],\ [w_2, 2],\ \ldots\ ,\ [w_n, n]\}$.

Once we have identified the beginning of the repair region, we introduce a *modification* to $w_{1..n} \in \Sigma^+$, $M(w)$, as a series of edit operations, $\{E_i\}_{i=1}^n$, in which each E_i is applied to w_i and possibly consists of a sequences of insertions before w_i, replacement or deletion of w_i, or transposition with w_{i+1}. This topological structure can be used to restrict the notion of modification, looking for conditions that guarantee the ability to recover the error. So, given $x_{1..m}$ a prefix in $\mathcal{L}(\mathcal{A})$, and $w \in \Sigma^+$, such that xw is not a prefix in $\mathcal{L}(\mathcal{A})$, we define a *repair of w following x* as $M(w)$, so that:

(1) $\mathcal{M}(q_0.x_{1..m}) = \mathcal{R}_{q_s}^{q_d}$ (the minimal region including the point of error, $x_{1..m}$)
(2) $\exists\{q_0.x_{1..i} = q_s.x_i, \ldots, q_s.x_{i..m}.M(w)\} \in \text{paths}(\mathcal{R}_{q_s}^{q_d})$

denoted by *repair*(x,w), and $\mathcal{R}_{q_s}^{q_d}$ by *scope*(M). We can now organize this concept around a point of error, $y_i \in y_{1..n}$, in order to take into account all possible repair alternatives. So, we define the *set of repairs for y_i*, as *repair*$(y_i) = \{xM(w) \in \text{repair}(x,w)/w_1 = \text{detection}(y_i)\}$.

Later, we focus on filter out undesirable repairs, introducing criteria to select those with minimal cost. For each $a, b \in \Sigma$ we assume insert, $I(a)$; delete, $D(a)$, replace, $R(a,b)$, and transpose, $T(a,b)$, costs. The *cost of a modification* $M(w_{1..n})$ is given by *cost*$(M(w_{1..n})) = \Sigma_{j\in J_\dashv} I(a_j) + \Sigma_{i=1}^n(\Sigma_{j\in J_i} I(a_j) + D(w_i) + R(w_i,b) + T(w_i,w_{i+1}))$, where $\{a_j,\ j \in J_i\}$ is the set of insertions applied before w_i; $w_{n+1} = \dashv$ the end of the input and $T_{w_n,\dashv} = 0$. From this, we define the set of *regional repairs* for $y_i \in y_{1..n}$, a point of error, as

$$\text{regional}(y_i) = \{xM(w) \in \text{repair}(y_i) \Big/ \begin{array}{l} \text{cost}(M) \le \text{cost}(M'),\ \forall M' \in \text{repair}(x,w) \\ \text{cost}(M) = \min_{L\in\text{repair}(y_i)}\{\text{cost}(L)\} \end{array} \}$$

Before to deal with cascaded errors, precipitated by previous erroneous repairs, it is necessary to establish the relationship between recovery processes. So, given w_i and w_j points of error, $j > i$, we define the set of *viable repairs* for w_i in w_j as $\text{viable}(w_i, w_j) = \{xM(y) \in \text{regional}(w_i)/xM(y)\ldots w_j \text{ prefix for } \mathcal{L}(\mathcal{A})\}$. Repairs in *viable*$(w_i, w_j)$ are the only ones capable of ensuring the recognition in $w_{i..j}$ and, therefore, the only possible at the origin of cascaded errors. In this sense, we say that a point of error $w_k,\ k > j$ is a *point of error precipitated by w_j* iff $\forall xM(y) \in \text{viable}(w_j, w_k),\ \exists \mathcal{R}_{q_0.w_{1..i}}^{q_d}$ defining $w_i = \text{detection}(w_j)$, such that $\text{scope}(M) \subset \mathcal{R}_{q_0.w_{1..i}}^{q_d}$. This implies that w_k is precipitated by w_j when the region defining the point of detection for w_k summarizes all viable repairs for w_j in w_k. That is, the information compiled from those repairs has not been sufficient to give continuity to a process locating the new error in a region containing the precedent ones and, as a consequence, depending on these. We then conclude that the origin of the current error could be a wrong study of past ones.

2.3 The Algorithm

Most authors appeal to global methods to avoid distortions due to unsafe error location [5, 6]; but our proposal applies a dynamic estimation of the repair region, guided by the linguistic knowledge present in the underlying FA. Formally, we extend the item structure, $[p, i, e]$, where now e is the error counter accumulated in the recognition of w at position w_i in state p.

Once located the point of error, we apply all possible transitions beginning at its point of detection, which corresponds to the following deduction steps in error mode, $\mathcal{D}_{\text{error}} = \mathcal{D}^{\text{Shift}}_{\text{error}} \cup \mathcal{D}^{\text{Insert}}_{\text{error}} \cup \mathcal{D}^{\text{Delete}}_{\text{error}} \cup \mathcal{D}^{\text{Replace}}_{\text{error}} \cup \mathcal{D}^{\text{Transpose}}_{\text{error}}$:

$$\begin{aligned}
\mathcal{D}^{\text{Shift}}_{\text{error}} &= \{[p,i,e] \vdash [q,i+1,e],\ \exists [a,i] \in \mathcal{H},\ q = p.a\} \\
\mathcal{D}^{\text{Insert}}_{\text{error}} &= \{[p,i,e] \vdash [p,i+1,e+I(a)],\ \not\exists\, p.a\} \\
\mathcal{D}^{\text{Delete}}_{\text{error}} &= \{[p,i,e] \vdash [q,i-1,e+D(w_i)] \Big/ \begin{array}{l} \mathcal{M}(q_0.w_{1..j}) = \mathcal{R}^{q_d}_{q_s} \\ p.w_i = q_d \in \mathcal{R}^{q_d}_{q_s} \textbf{or } q = q_d \end{array} \} \\
\mathcal{D}^{\text{Replace}}_{\text{error}} &= \{[p,i,e] \vdash [q,i+1,e+R(w_i,a)], \Big/ \begin{array}{l} \mathcal{M}(q_0.w_{1..j}) = \mathcal{R}^{q_d}_{q_s} \\ p.a = q \in \mathcal{R}^{q_d}_{q_s} \textbf{ or } q = q_d \end{array} \} \\
\mathcal{D}^{\text{Transpose}}_{\text{error}} &= \{[p,i,e] \vdash [q,i+2,e+T(w_i,w_{i+1})] \Big/ \begin{array}{l} \mathcal{M}(q_0.w_{1..j}) = \mathcal{R}^{q_d}_{q_s} \\ p.w_i.w_{i+1} = q \in \mathcal{R}^{q_d}_{q_s} \textbf{ or } q = q_d \end{array} \}
\end{aligned}$$

where $w_{1..j}$ looks for the current point of error. Observe that, in any case, the error hypotheses apply on transitions behind the repair region. The process continues until a repair covers the repair region.

In the case of dealing with an error which is not the first one in the word, it could condition a previous repair. This arises when we realize that we come back to a detection item for which some recognition branch includes a previous recovery process. The algorithm re-takes the error counters, adding the cost of new error hypotheses to profit from the experience gained from previous repairs. This permits us to deduce that if w_l is a point of error precipitated by w_k, then:

$$q_0.w_{1..i} < q_0.w_{1..j},\ \mathcal{M}(q_0.w_l) = \mathcal{R}^{q_d}_{q_0.w_{1..i}},\ w_j = y_1,\ xM(y) \in \text{viable}(w_k, w_l)$$

which proves that the state associated to the point of detection in a cascaded error is minor that the one associated to the source of the scope in the repairs precipitating it. So, the minimal possible scope of a repair for the cascaded error includes any scope of the previous ones, that is,

$$\max\{\mathit{scope}(M),\ M \in \text{viable}(w_k, w_l)\} \subset \max\{\mathit{scope}(\tilde{M}),\ \tilde{M} \in \text{regional}(w_l)\}$$

This allows us to get an asymptotic behavior close to global repair methods, ensuring a quality comparable to those, but at cost of a local one in practice.

3 An Overview on Galician

Although Galician is a non-agglutinative language, it shows a great variety of morphological processes. The most outstanding features are found in verbs, with a highly complex conjugation paradigm, including ten simple tenses. If we add the present imperative with two forms, not conjugated infinitive, gerund and participle. Then 65 inflected forms are possible for each verb. In addition, irregularities are present in both stems and endings. So, very common verbs, such as `facer` (*to do*), have up to five different stems: `fac-er`, `fag-o`, `fa-s`, `fac-emos`, `fix-en`. Approximately 30% of Galician verbs are irregular. We have implemented 42 groups of irregular verbs. Verbs also include enclitic pronouns producing changes in the stem due to the presence of accents: `deu` (*gave*), `déullelo` (*he/she gave it to them*).

In Galician the unstressed pronouns are usually suffixed and, moreover, pronouns can be easily drawn together and they can also be contracted (`lle + o` = `llo`), as in the case of `váitemello buscar` (*go and fetch it for him (do it for me)*). It is also very common to use what we call a *solidarity pronoun*, in order to let the listeners be participant in the action. Therefore, we have even implemented forms with four enclitic pronouns, like `perdéuchellevolo` (*he had lost it to him*). Here, the pronouns `che` and `vos` are solidarity pronouns and they are used to implicate the interlocutor in the facts that are being told. None of them has a translation into English, because this language lacks these kinds of pronouns. So, the analysis has to segment the word and return five tokens.

There exist highly irregular verbs that cannot be classified in any irregular model, such as `ir` (*to go*) or `ser` (*to be*); and other verbs include gaps in which some forms are missing or simply not used. For instance, meteorological verbs such as `chover` (*to rain*) are conjugated only in third singular person. Finally, verbs can present duplicate past participles, like `impreso` and `imprimido` (*printed*).

This complexity extends to gender inflection, with words with only one gender as `home` (*man*) and `muller` (*woman*), and words with the same form for both genders as `azul` (*blue*). We also have a lot of models for words with separate masculine and feminine forms: `autor`, `autora` (*author*); `xefe`, `xefa` (*boss*); `poeta`, `poetisa` (*poet*); `rei`, `raiña` (*king*) or `actor`, `actriz` (*actor*). We have implemented 33 variation groups for gender.

We can also refer to number inflection, with words only being presented in singular form, such as `luns` (*monday*), and others where only the plural form is correct, as `matemáticas` (*mathematics*). The construction of different forms does not involve as many variants as is the case for gender, but we can also consider a certain number of models: `roxo`, `roxos` (*red*); `luz`, `luces` (*light*); `animal`, `animais` (*animal*); `inglés`, `ingleses` (*English*); `azul`, `azuis` (*blue*) or `funil`, `funís` (*funnel*). We have implemented 13 variation groups for number.

4 The System at Work

Our aim is to validate our proposal comparing it with global ones, an objective criterion to measure the quality of a repair algorithm since the point of reference is a technique that guarantees the best quality for a given error metric when all contextual information is available. We choose to work with a lexicon for Galician built from Galena [3], which includes 304.331 different words, to illustrate this aspect. The lexicon is recognized by a FA containing 16.837 states connected by 43.446 transitions, whose entity we consider sufficient for our purposes.

4.1 The Operational Testing Frame

From this lexicon, we select a representative sample of morphological errors to its practical evaluation. This can be easily verified from Fig. 1, that shows the similar distribution of both the original lexicon and the running sample, in terms of lengths of the words to deal with. In each length-category, errors have been randomly generated in a number and position in the input string that are shown in Fig. 2. This is of importance since, as the authors claim, the performance of previous proposals depend on these factors, which has no practical sense. No other dependencies have been detected at morphological level and, therefore, they have not been considered.

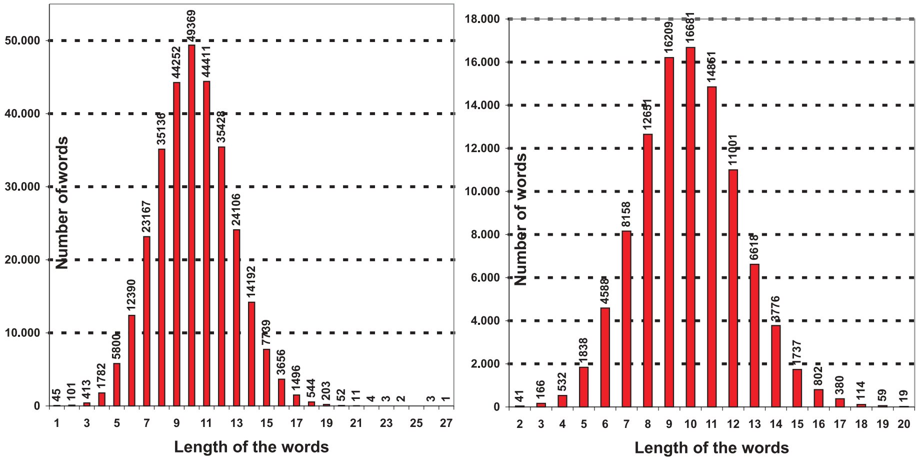

Fig. 1. Statistics on the general and error lexicons

In this context, our testing framework seems to be well balanced, from both viewpoints operational and linguistic. It remains to decide what repair algorithms will be tested. We compare our proposal with the Savary's global approach [6], an evolution of the Oflazer's algorithm [5] and, in the best of our knowledge, the most efficient method of error-tolerant look-up in finite-state dictionaries. The comparison has been done from three viewpoints: the size of the repair region considered, the computational cost and the repair quality.

4.2 The Error Repair Region

We focus on the evolution of this region in relation to the location of the point of error, in opposition to static strategies associated to global repair approaches. To illustrate it, we take as running example the FA represented in Fig. 3, which recognizes the following words in Galician: *"chourizo"* (sausage), *"cohabitante"* (a person who cohabit with another one), *"coherente"* (coherent) and *"cooperase"* (you cooperated). We consider as input string the erroneous one *"coharizo"*, resulting from transpose *"h"* with *"o"* in *"chourizo"* (sausage), and replace the character *"u"* by *"a"*. We shall describe the behavior from both viewpoints, the Savary's [6] algorithm and our proposal, proving that in the worst case, when precipitated errors are present, our proposal can re-take the repair process to recover the system from errors in cascade.

In this context, the recognition comes to an halt on state q_9, for which $\mathcal{M}(q_9) = \mathcal{R}_{q_6}^{q_{22}}$ and no transition is possible on *"r"*. So, our approach locates the error at q_6 and applies from it the error hypotheses looking for the minor editing distance in a repair allowing to reach the state q_{22}. In this case, there are two possible regional repairs consisting on first replace *"a"* by *"e"* and later insert an *"e"* after *"r"* (resp. replace *"i"* by *"e"*), to obtain the modification on the entire input string *"coherezo"* (resp. *"cohereizo"*), which is not a word in our running language.

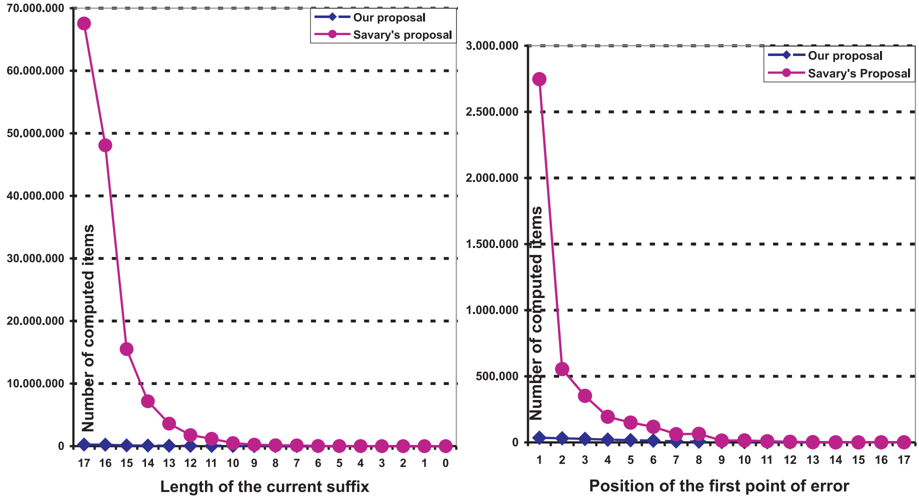

Fig. 2. Number of items generated in error mode

As a consequence, although we return to the standard recognition in q_{22}, the next input character is now *"i"* (resp. *"z"*), for which no transition is possible and we come back to error mode on the region $\mathcal{M}(q_{22}) = \mathcal{R}_{q_4}^{q_{24}}$ including $\mathcal{M}(q_9) = \mathcal{R}_{q_6}^{q_{22}}$. We then interpret that the current error is precipitated by the previous one, possibly of type in cascade. As result, none of the regional repairs generated allow us to re-take the standard recognition beyond the state q_{24}. At this point, $\mathcal{M}(q_{24}) = \mathcal{R}_{q_2}^{q_{25}}$ becomes the new region, and the only regional repair is now defined as the transposition of the *"h"* with *"o"*, and the substitution of *"a"* by *"u"*; which agrees with the global repair proposed by Savary, although the repair region is not the total one as is the case for that algorithm. This repair finally allows the acceptance by the FA.

The repair process described is interesting for two reasons. First, it puts into evidence that we do not need to extend the repair region to the entire FA in order to get the least-cost correction and, secondly, the risk of errors in cascade can be efficiently solved in the context of non-global approaches. Finally, in the worst case, our running example clearly illustrate the convergence of our regional strategy towards the global one from both viewpoints, the computational cost and the quality of the correction.

4.3 The Computational Cost

These practical results are compiled in Fig. 2, using as unity to measure the computational effort the concept of item previously defined. We here consider two complementary approaches illustrating the dependence on both the position of the first point of error in the word and the length of the suffix from it. So, in any case, we are sure to take into account the degree of penetration in the FA at that point, which determines the effectiveness of the repair strategy. In effect, working on regional methods, the penetration determines the number of regions in the FA including the point of error and, as a consequence, the possibility to consider a non-global resolution.

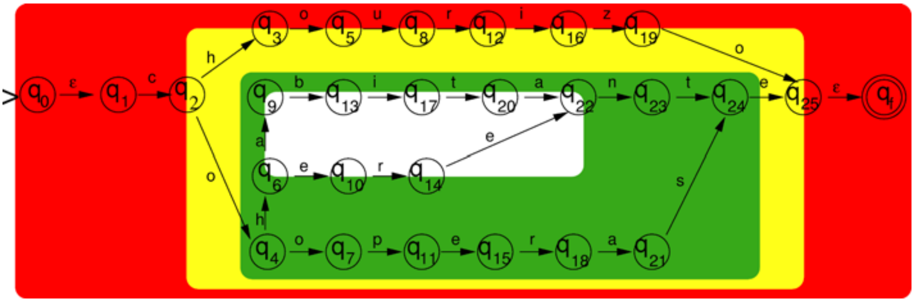

Fig. 3. The concept of region in error repair

In order to clearly show the detail of the tests on errors located at the end of the word, which is not easy to observe from the decimal scale of Fig. 2, we include in Fig. 4 the same results using a logarithmic scale. So, both graphics perfectly illustrate our contribution, in terms of computational effort saved, from two viewpoints which are of interest in real systems: First, our proposal shows in practice a linear-like behavior, in opposite to the Savary's one that seems to be of exponential type. In particular, this translates in an essential property in industrial applications, the independence of the the time of response on the initial conditions for the repair process. Second, in any case, the number of computations is significantly reduced when we apply our regional criterion.

4.4 The Performance

However, statistics on computational cost only provide a partial view of the repair process that must also take into account data related to the performance from both the user's and the system's viewpoint. In order to get this, we have introduced the following two measures, for a given word, w, containing an error:

$$performance(w) = \frac{useful\ items}{total\ items} \qquad recall(w) = \frac{proposed\ corrections}{total\ corrections}$$

that we complement with a global measure on the *precision* of the error repair approach in each case, that is, the rate reflecting when the algorithm provides the correction attended by the user. We use the term *useful items* to refer to the number of generated items that finally contribute to obtain a repair, and *total items* to refer to the number of these structures generated during the process. We denote by *proposed corrections* the number of corrections provided by the algorithm, and by *total corrections* the number of possible ones, absolutely.

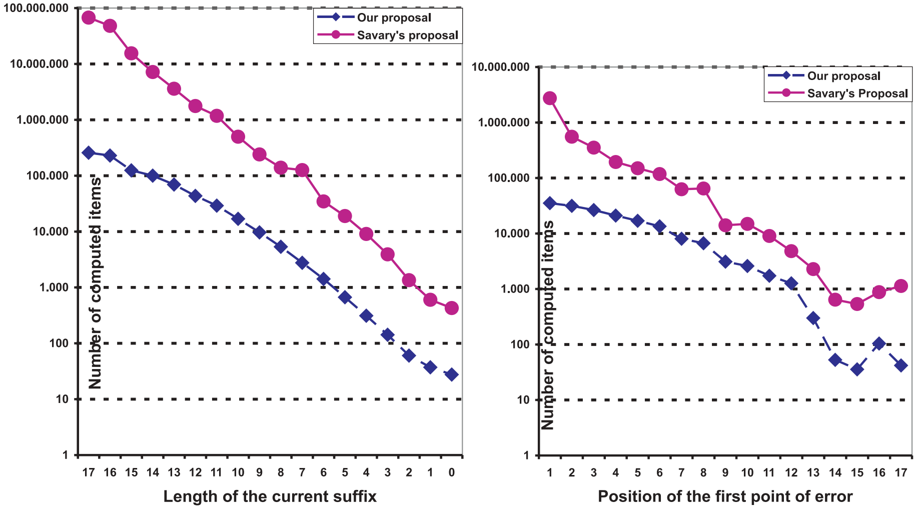

Fig. 4. Number of items generated in error mode. Logarithmic scale

These results are shown in Fig. 5, illustrating some interesting aspects in relation with the asymptotic behavior we want to put into evidence in the regional approach. So, considering the running example, the performance in our case is not only better than Savary's; but the existing difference between them increases with the location of the first point of error. Intuitively this is due to the fact that closer is this point to the beginning of the word and greater is the number of useless items generated in error mode, a simple consequence of the higher availability of different repair paths in the FA when we are working in a region close to q_0. In effect, given that the concept of region is associated to the definition of corresponding source and drain points, this implies that this kind of regions are often equivalent to the total one since the disposition of these regions is always concentric. At this point, regional and repair approaches apply the same error hypotheses not only on a same region, but also from close states given that, in any case, one of the starting points for these hypotheses would be q_0 or a state close to it. That is, in the worst case, both algorithms converge.

The same reasoning could be considered in relation to points of error associated to a state in the recognition that is close to q_f, in order to estimate the repair region. However, in this case, the number of items generated is greater for the global technique, which is due to the fact that the morphology of the language often results on the generation of regions which concentrate near of q_f, a simple consequence of the common derivative mechanisms applied on suffixes defining gender, number or verbal conjugation groups. So, it is possible to find a regional repair just implicating some error hypotheses from the state associated to the point of error or from the associated detection point and, although this regional repair could be different of the global one; its computational cost would be usually minor.

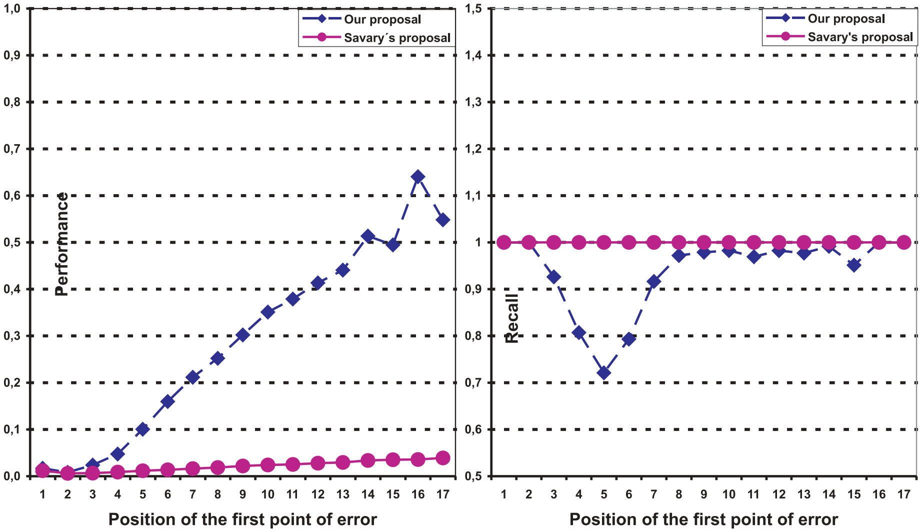

Fig. 5. Performance and recall results

A similar behavior can be observed with respect to the recall relation. Here, Savary's algorithm shows a constant graph since the approach applied is global and, as consequence, the set of corrections provided is always the entire one for a fixed error counter. In our proposal, the results prove that the recall is smaller than for Savary's, which illustrates the gain in computational efficiency in opposite to the global method. Related to the convergence between regional and global approaches, we must again search around points of detection close to the beginning of the word, which often also implies repair regions be equivalent to the total one and repairs starting around of q_0, such as is illustrated in Fig. 5.

However, in opposite to the case of performance, we remark that for recall the convergence between global and regional proposals seems also extend to processes where the point of error is associated to states close to q_f, that is, when this point is located near of the end of the word. To understand this, it is sufficient to take into account that we are not now computing the number of items generated in the repair, but the number of corrections finally proposed. So, given that closer to the end of the word we are and smaller is the number of alternatives for a repair process, both global and regional approaches converge also towards the right of the graph for recall.

Finally, the regional (resp. the global) approach provided as correction the word from which the error was randomly included in a 77% (resp. 81%) of the cases. Although this could be interpreted as a justification to use global methods, it is necessary to remember that we are now only taking into account morphological information, which has an impact in the precision for a regional approach, but not for a global one that always provide all the repair alternatives without exclusion. So, the precision represents, in an exclusively morphological context, a disadvantage for our proposal since we base the efficiency in the limitation of the search space. The future integration of linguistic information from both, syntactic and semantic viewpoints should reduce significantly this gap, less than 4%, around the precision; or even should eliminate it.

5 Conclusion

The design of computational tools for linguistic usage should respond to the constraints of efficiency, safety and maintenance. So, a major point of interest in dealing with these aspects is the development of error correction strategies, since this kind of techniques supplies the robustness necessary to extend formal prototypes to practical applications. In this paper, we have described a proposal on spelling correction for Galician, a Latin language with non-trivial morphology trying to rescue its recognition from society, which involves to have tools in order to ensure a correct usage of it. We take advantage of the grammatical structure present in the underlying morphological recognizer to provide the user an automatic assistant to develop linguistic tasks without errors. In this sense, our work represents an initial approach to the problem, but preliminary results seem to be promising and the formalism well adapted to deal with more complex problems such as the consideration of additional linguistic knowledge.

References

1. J. Daciuk, S. Mihov, B.W. Watson, and R.E. Watson. Incremental construction of minimal acyclic finite-state automata. *Computational Linguistics*, 26(1):3–16, 2000.
2. M. W. Du and S.C. Chang. A model and a fast algorithm for multiple errors spelling correction. *Acta Informatica*, 29(3):281–302, June 1992.

3. J. Graña, F.M. Barcala, and M.A. Alonso. Compilation methods of minimal acyclic automata for large dictionaries. *Lecture Notes in Computer Science*, 2494:135–148, 2002.
4. C.L. Lucchesi and T. Kowaltowski. Applications of finite automata representing large vocabularies. *Software-Practice and Experience*, 23(1):15–30, January 1993.
5. K. Oflazer. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics*, 22(1):73–89, 1996.
6. A. Savary. Typographical nearest-neighbor search in a finite-state lexicon and its application to spelling correction. *Lecture Notes in Computer Science*, 2494:251–260, 2001.
7. K. Sikkel. *Parsing Schemata*. PhD thesis, Univ. of Twente, The Netherlands, 1993.