Spelling Correction on Technical Documents[*]

M. Vilares[1], J. Otero[1], and J. Graña[2]

[1] Department of Computer Science, University of Vigo
Campus As Lagoas s/n, 32004 Ourense, Spain
{vilares,jop}@uvigo.es
[2] Department of Computer Science, University of A Coruña
Campus de Elviña s/n, 15071 A Coruña, Spain
grana@udc.es

Abstract. We describe a novel approach to spelling correction applied on technical documents, a task that requires a number of especific properties such as efficiency, safety and maintenance. In opposite to previous works, we explore the region close to the point at which the recognition halts, gathering all relevant information for the repair process in order to avoid the phenomenom of errors in cascade. Our approach seems to reach the same quality provided by the most performance classic techniques, but with a significant reduction on both time and space costs.

1 Introduction

Although much effort has gone into the problem of spelling error correction over the years, the devising estrategies remains a research challenge, and even there is a renewal interest on it due to the increasing amount of available information in electronic format or placed on line. In particular, spelling correction is a critical task in dealing with the elaboration of technical documents, for which efficiency, safety and maintenance are properties that cannot be negligented.

In opposite to fully automatic correction [4], most commercial systems assist users by offering a set of candidate corrections that are close to misspelled word. This allows to avoid wrong interpretations in a field where the meaning of a word cannot often be approximately defined and the semantic relations in the vocabulary are very strict. At this point, any technique allowing the repair process to reduce the number of candidates considered for correction translates in an improved efficiency and maintenance, that should not have side effects on safety. In order to get this last, we focus on limit the size of the repair region in the word, in contrast to previous proposals considering a global one.

2 The operational 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

[*] Research supported by the Spanish Government under projects TIN2004-07246-C03-01, TIN2004-07246-C03-02 and HP2002-0081, and the Autonomous Government of Galicia under projects PGIDIT03SIN30501PR and PGIDIT02SIN01E.

w_1 (resp. following w_n). We generate from $\mathcal{G}$ a *numbered minimal acyclic finite automaton* for the language $\mathcal{L}(\mathcal{G})$. A *finite automaton* (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 $(\overset{n-2}{\ldots}\ (q.w_1)\ \overset{n-2}{\ldots}).w_n$. 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 [2]. In this sense, we say that 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. Although the standard recognition 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 triplet $\langle \mathcal{I}, \mathcal{H}, \mathcal{D} \rangle$, 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_p^w, 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.

3 Spelling correction

We talk about the *error* in a portion of the word to mean the difference between what was intended and what actually appears in the word. So, we can talk about the *point of error* as the point at which the difference occurs.

In this context, a *repair* should be understood as a modification on the input string allowing the recognizer both, to recover the standard process and to avoid

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

the phenomenom of cascaded errors, that is, errors precipitated by a previous erroneous repair diagnostic. That is, precisely, the goal of the notion of *regional repair* defined in [8], which we succintly introduce now.

Given that we work with acyclic FAs, we can consider a simple order relation between two states, p and q, in such a way that $p < q$ iff exists a path in the FA from p to q. We say that a pair of states (p, q) is a *region* in the FA when it defines a sub-automaton with initial (resp. final) state in p (resp. q). So, we say that a state r is in the region defined by the pair (p, q), denotated by $\mathcal{R}_p^q$, iff there exists a path ρ in $\mathcal{R}_p^q$, such that $r \in \rho$. Given $r \in \mathcal{Q}$, it can be proved that there exists an only *minimal region* in the FA containing it.

To begin with, we assume that we are dealing with the first error detected. We extend the initial estructure of items, as a pair $[p, i]$, with an error counter e; resulting in a new estructure of the form $[p, i, e]$. For a given *point of error*, w_j, we locate the associated *point of detection*, defined as the source of the minimal region, $\mathcal{M}(w_j) = \mathcal{R}_p^q$, containing w_j. Associated to the point of error, w_j, (resp. point of detection, w_i) we consider the corresponding *error* (resp. *detection*) *item* iff is of the form $[q, j]$ (resp. $[p, i]$). To filter out undesirable repairs, we introduce 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. Once the detection item has been fixed, we apply from it the deduction steps $\mathcal{D}_{\text{error}} = \mathcal{D}^{\text{Shift}} \cup \mathcal{D}_{\text{error}}^{\text{Insert}} \cup \mathcal{D}_{\text{error}}^{\text{Delete}} \cup \mathcal{D}_{\text{error}}^{\text{Replace}} \cup \mathcal{D}_{\text{error}}^{\text{Transpose}}$, defined as follows:

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

$$\mathcal{D}_{\text{error}}^{\text{Insert}} = \{[p, i, e] \vdash [p, i+1, e + I(a)]\}$$

$$\mathcal{D}_{\text{error}}^{\text{Delete}} = \{[p, i, e] \vdash [q, i-1, e + D(w_i)] \Big/ \begin{array}{l} \mathcal{M}(q_0.w_{1..j}) = \mathcal{R}_{q_s}^{q_d} \\ p.w_i = q_d \in \mathcal{R}_{q_s}^{q_d} \textbf{or } q = q_d \end{array} \}$$

$$\mathcal{D}_{\text{error}}^{\text{Replace}} = \{[p, i, e] \vdash [q, i+1, e + R(w_i, a)], \Big/ \begin{array}{l} \mathcal{M}(q_0.w_{1..j}) = \mathcal{R}_{q_s}^{q_d} \\ p.a = q \in \mathcal{R}_{q_s}^{q_d} \textbf{ or } q = q_d \end{array} \}$$

$$\mathcal{D}_{\text{error}}^{\text{Transpose}} = \{[p, i, e] \vdash [q, i+2, e + T(w_i, w_{i+1})] \Big/ \begin{array}{l} \mathcal{M}(q_0.w_{1..j}) = \mathcal{R}_{q_s}^{q_d} \\ p.w_i.w_{i+1} = q \in \mathcal{R}_{q_s}^{q_d} \textbf{ or } q = q_d \end{array} \}$$

where $w_{1..j}$ looks for the current point of error. Note that, in any case, the error hypotheses apply on transitions behind the repair region. The process continues until a repair covers that region, accepting a character in the remaining string. To avoid the generation of items only differenciated by the error counter, we apply a principle of optimization, saving only for computation purposes those with minimal counters.

When the current repair is not the first one, it can modify a previous repair in order to avoid cascaded errors, by adding the cost of the new error hypotheses to profit from the experience gained from previous ones. This allows us to get, in a simple manner, an asymptotic behavior close to global repair methods [8]. The time complexity is, in the worst case

$$\mathcal{O}(\frac{n!}{\tau! * (n-\tau)!} * (n+\tau) * 2^{\tau} * \text{fan-out}_{\mu}^{\tau})$$

where τ and $fan\text{-}out_\mu$ are, respectively, the maximal error counter computed and the maximal fan-out of the automaton in the scope of the repairs considered. The input string is recognized iff a final item $[q_f, n+1, e]$, $q_f \in \mathcal{Q}_f$, is generated.

4 The experimental frame

Our aim now is to validate the practical interest of our proposal in relation to classic global repair strategies. We think that this is an objective criterion since the point of reference is a technique that guarantees the best quality for a given error metric when all contextual information is available.

4.1 The running language

Our running language is Spanish, with a great variety of morphological processes, making it adequate for our description. The most outstanding features are to be found in verbs, with a highly complex conjugation paradigm, including nine simple tenses and nine compound tenses, all of which have six different persons. If we add the present imperative with two forms, the infinitive, the compound infinitive, the gerund, the compound gerund, and the participle with four forms, then 118 inflected forms are possible for each verb. In addition, irregularities are present in both stems and endings. So, very common verbs, such as `hacer` (*to do*), have up to seven different stems: `hac-er`, `hag-o`, `hic-e`, `har-é`, `hiz-o`, `haz`, `hech-o`. Approximately 30% of Spanish verbs are irregular, and can be grouped around 38 different models. Verbs also include enclitic pronouns producing changes in the stem due to the presence of accents: `da` (*give*), `dame` (*give me*), `dámelo` (*give it to me*). We have considered forms with up to three enclitic pronouns, like `tráetemelo` (*bring it for you and me*). There exist some highly irregular verbs that cannot be classified in any irregular model, such as `ir` (*to go*) or `ser` (*to be*); and others include gaps in which some forms are missing or simply not used. For instance, meteorological verbs such as `nevar` (*to snow*) are conjugated only in third person singular. Finally, verbs can present duplicate past participles, like `impreso` and `imprimido` (*printed*).

This complexity extends to gender inflection, with words considering only one gender, such as `hombre` (*man*) and `mujer` (*woman*), and words with the same form for both genders, such as `azul` (*blue*). In relation to words with separate forms for masculine and feminine, we have a lot of models: `autor`, `autora` (*author*); `jefe`, `jefa` (*boss*); `poeta`, `poetisa` (*poet*); `rey`, `reina` (*king*) or `actor`, `actriz` (*actor*). We have considered 20 variation groups for gender. We can also refer to number inflection, with words presenting only the singular form, as `estrés` (*stress*), and others where only the plural form is correct, as `matemáticas` (*mathematics*). The construction of different forms does not involve as many variants as in the case of gender, but we can also consider a certain number of models: `rojo`, `rojos` (*red*); `luz`, `luces` (*light*); `lord`, `lores` (*lord*) or `frac`, `fraques` (*dress coat*). We have considered 10 variation groups for number.

4.2 The corpus

We choose to work with the Itu corpus[2], the main collection of texts about telecommunications. In particular, we have considered a sub-corpus of 17.423 words that have been used as a part of the support for the Crater project [1]. The recognizer is a *finite automaton* (fa) containing 11.193 states connected by 23.278 transitions built from Galena [3]. The distribution, in terms of lengths of the words dealt with, is shown in Fig. 1.

For each length-category, errors have been randomly generated in a number and position for the first error 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, for example in terms of lexical categories, have been detected at morphological level and, therefore, they have not been considered.

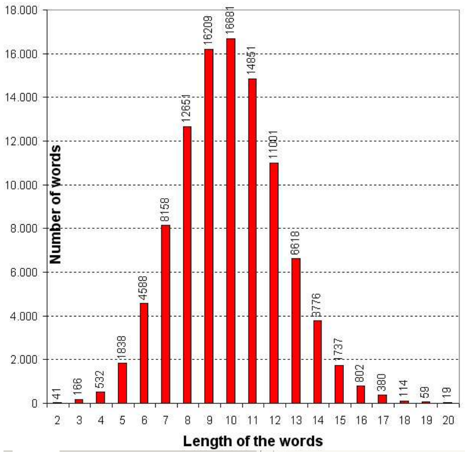

Fig. 1. Statistics on the lexicon.

In this context, our testing framework seems to be well balanced, from both viewpoints operational and linguistic, in order to estimate the practical performance of error repair algorithms on fa architectures. It only remains to decide what repair algorithms will be tested. We choose to 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 complementary viewpoints: the size of the repair region considered, the computational cost and the quality exhibed.

[2] For *International Telecommunications Union CCITT Handbook.*

4.3 The computational cost

Practical results are compiled in Fig. 2, using as unity to measure the computational effort the concept of item previously defined. In order to take the *edit distance* [5] as the error metric for measuring the quality of a repair, it is sufficient to consider discrete costs $I(a) = D(a) = 1, \forall a \in \Sigma$ and $R(a,b) = T(a,b) = 1, \forall a,b \in \Sigma, a \neq b$. 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 estrategy. 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.

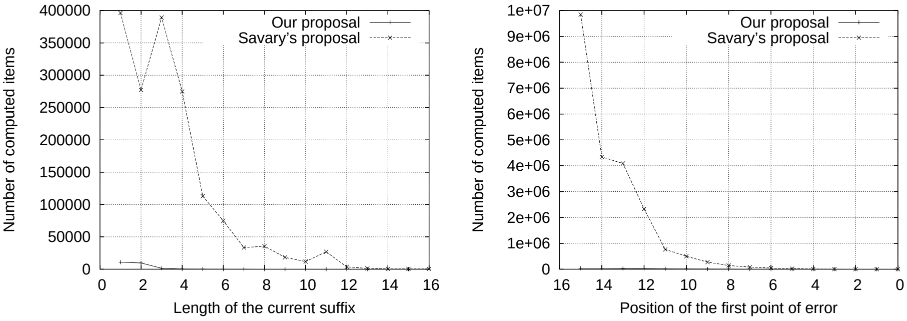

Fig. 2. Number of items generated in error mode.

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. 3 the same results using a logarithmic scale. So, both graphics perfectly illustrate our apportation, 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.

These results are shown in Fig. 4, 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.

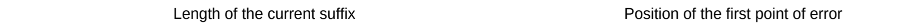

Fig. 3. Number of items generated in error mode. Logarithmic scale.

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 derivational 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.

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. 4.

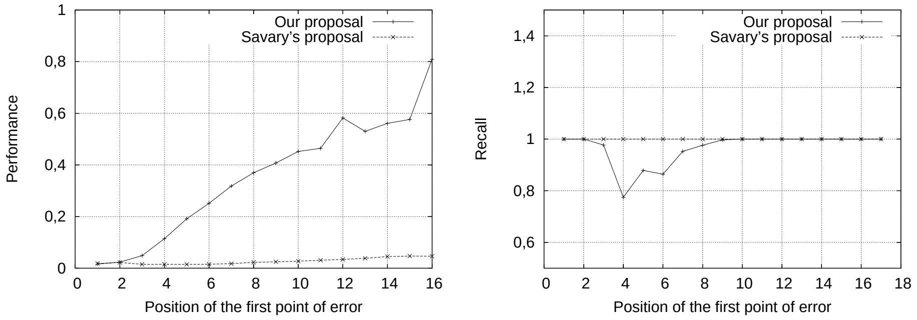

Fig. 4. Performance and recall results.

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 consideration of the precision concept represents, in the exclusive morphological context considered, a clear disadvantage for our proposal since it bases its efficiency in the limitation of the search space. We attend that the integration of linguistic information from both, syntactic and semantic viewpoints will reduce significantly this gap, less than 4%, around the precision; or even will eliminate it.

5 Conclusion

We have illustrated how a least-cost regional error repair technique can be applied to the task of spelling correction on technical texts, a field where isolated-word estrategies can be preferable to context-sensitive ones and, as a consequence, morphological aspects strongly impact the performance.

Our proposal adapts to any FA-based frame and no particular requeriments are needed to put it into running. The system was evaluated against a set of texts artificially generated from the ITU corpus in telecommunications, and preliminary results seem to indicate that the system can be used for removing many of the lexical errors in the input of a technical document.

References

1. Thorsten Brants. Some experiments with the CRATER corpus. Technical report, Universität des Saarlandes, Saarbrücken, 1995.
2. 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.
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. K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439, December 1992.
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.
8. M. Vilares, J. Otero, and J. Graña. Regional finite-state error repair. In *Proc. of the Ninth Int. Conf. on Implementation and Application of Automata (CIAA'04)*, Kingston, Canada, 2004.