UNIVERSITY OF SÃO PAULO

School of Arts, Sciences and Humanities

Technical Report PPgSI-001/2014

An OO Writer Module for Spelling Correction in Brazilian Portuguese

Priscila Azar Gimenes
Norton Trevisan Roman
Ariadne Maria Brito Rizzoni Carvalho

July - 2014

Technical Report Series

PPgSI-EACH-USP. Arlindo Béttio St. 1000 - Ermelino Matarazzo - 03828-000.
São Paulo, SP. Brazil.
TEL: 55 (11) 3091-8197 http://www.each.usp.br/ppgsi

An OO Writer Module for Spelling Correction in Brazilian Portuguese

Priscila A. Gimenes[1] **, Norton T. Roman**[1] **, Ariadne M. B. R. Carvalho**[2]

[1]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo
São Paulo – SP, Brazil

{priscila.gimenes,norton}@usp.br

[2]Instituto de Computação – Universidade Estadual de Campinas
Campinas – SP, Brazil

ariadne@ic.unicamp.br

Abstract. *This technical report presents an OO Writer module for rearranging the spelling suggestion list in Brazilian Portuguese. To do so, the module relies on some statistics collected from texts typed in this language. As it turned out, a comparison between the lists generated by the newly added module and the ones originally suggested by Open Office Writer showed an improvement regarding the order in which suggestions are presented to the user.*

1. Introduction

When typing a text, several spelling mistakes may occur [Medeiros 1995]. To deal with this problem, many spelling checkers have been developed over the years. They are usually embedded in applications, such as email services and text editors. Spelling correction usually means either correcting a misspelled word automatically, or presenting a list of suggestions to the user, or both. During the analysis, at least two different types of mistakes can be found: (1) non-word errors, that is, words that simply do not exist; and (2) real-word errors, that is, words that do exist but which do not make sense in the current context [Kukich 1992]. It is very difficult do deal with errors of the second type, because their correction may depend on a deeper analysis of the text [Kukich 1992].

Commercial spelling checkers usually focus on (1), but the order in which the list of suggested words is presented to the user may be a problem. For example, when the word `Nao` is typed in OpenOffice Writer[1] (OO Writer), the correct word `Não`[2] is suggested in sixth place in the list, whereas the word `Ao` comes on top. Another example is the word `Ve-lo`, whose correct form `Vê-lo`[3], when working with Microsoft Office Word[4](MS Word), does not even show up in the suggestion list.

The reason for this behaviour may be that, in general, statistics on which words should be considered for suggestion, and in which order, are derived from English texts (*cf.* [Damerau 1964]). Still, Medeiros [1995] showed that 80.1% of all spelling errors in European Portuguese would fit into one of Damerau's categories, to wit, **Insertion** (an extra letter is inserted), **Omission** (one letter is missing), **Substitution** (one letter is wrong)

[1]OPENOFFICE.ORG – The Free and Open Productivity Suite `http://openoffice.org`.
[2]"No" in Portuguese.
[3]Either "See it" or "See him" in Portuguese.
[4]Microsoft Office – Office.com `http://office.microsoft.com/pt-br`.

and **Transposition** (two adjacent characters were transposed). He, however, noticed that around 20% of all linguistic errors found in European Portuguese texts were related to the use of diacritics, showing that such marks should be taken into account when using string edit distance or probability models for spelling correction. Hence, these findings suggest that the ordering of suggestions should embody diacritical error information.

Starting from these observations, we have analysed 1,808 typed texts in Brazilian Portuguese [Roman et al. 2013], determining the frequency with which mistakes are produced. In the sequence, we built an extension to OpenOffice Writer to reorder the list of spelling suggestions offered to the user, according to these frequencies. In doing so, our intention was to provide an insight on the practical usefulness of these statistics alone, without having to rely on more elaborated procedures. The rest of the document is organised as follows. In Section 2 the main concepts are described; in Section 3 we explain the methodology used in the project; in Section 4 we discuss the analysis and the obtained results; finally, in Section 5 we present our conclusions and contributions.

2. Underlying Concepts

Spelling errors may happen due to several reasons, such as the author's lack of knowledge or mechanical problems during typing [Peterson 1986]. Thus, errors can be broadly divided in two categories: (1) orthographic, and (2) typographical errors (typos). The first are cognitive, and consist of typing the wrong word instead of the right one, usually because of their phonetic similarity [van Berkel and Smedt 1988]. Typographical errors, on the other hand, happen when a wrong sequence of characters is typed, usually depending on the used keyboard and the proximity of the keys. The language may also interfere in the order in which words are misspelled [van Berkel and Smedt 1988]. Alternatively, we can classify errors as (1) non-word errors, (which fit into one of Damerau's categories [Damerau 1964]), and (2) real-word errors [Kukich 1992], as previously explained.

Spelling correction, in turn, consists in verifying which words are incorrectly spelled, and then either correcting them automatically, or providing a list of suggestions to the user. A word may be analysed individually, or we can observe its surroundings to check, for instance, agreement and punctuation. A deeper analysis could also involve style and legibility of the text [Medeiros 1995]. Most spelling correctors deal with isolated words, ignoring context, thus failing when an orthographic error generates a correct word, for example [Deorowicz and Ciura 2005].

Among the several techniques for correcting isolated words, the most frequently used is edit distance [Jurafsky and Martin 2009, Deorowicz and Ciura 2005]. Edit distance is the minimal amount of editing operations needed to convert one string into another [Wagner and Fischer 1974]. For example, the edit distance between the words `amor`[5] and `amora`[6] is 1, because a single operation (inserting character `a`) would be needed to convert the first into the second. According to Damerau's work, the most common operations are: insertion, omission, transposition and substitution. In our work, we have incorporated a fifth operation to deal with accents.

Under this procedure, given some misspelled word, a dictionary is searched for some correctly-formed word at a minimum edit distance to the misspelled one. For example,

[5] "Love" in Portuguese.

[6] "Blackberry" in Portuguese.

to convert `Opnião` into the correct word `Opinião`[7], it is only necessary to insert the character `i` (therefore one operation only). Compared to errors, correcting operations take an opposite way: if the error is an omission, then an insertion is tried; if it is an insertion, an omission is tested; if it is a substitution, a new one is tried; finally, if it is a transposition, another transposition is done.

Those words from the dictionary requiring the smallest number of operations to be turned into the wrong word are taken as candidates for the correct form. Since the correction of isolated words may result in more than one suggestion, these must be ordered based on their likelihood, according to some pre-defined criterion, even if we cannot be completely sure that the most probable word is in fact the right one [Deorowicz and Ciura 2005].

Criteria for such ordering usually take the form of assigning costs to each type of error. Veronis [1988], for example, assigns the same cost (and therefore the same probability) to all operations and finds the amount of necessary operations to convert the misspelled word into a word in the dictionary. The words are then ranked, starting with the smallest edit distance, up to a specified bound. Alternatively, we may attribute different costs to different types of errors, so the edit distance is the sum of operations used in morphing one word into the other. In this case, each cost could represent, for instance, the probability of occurrence of a certain error (Wagner and Fischer [1974]).

3. Methodology

The main goal of this work was to implement an auxiliary module for OO Writer's Spelling correction facility, using statistics of errors occurring in Brazilian Portuguese, as an attempt to improve the order of suggestions in the list given by that system. To do so, we gathered error frequencies from a corpus of 1,808 typed texts in Brazilian Portuguese (C_1) [Roman et al. 2013], taking them as estimate probabilities to reorder the items in the spelling suggestion list. Next, an edit distance algorithm was chosen and tested with the corpus. The suggestion list generated by the algorithm was then compared to the original one suggested by OO writer. Our choice for OO writer was guided by the fact that this is an open source project, also providing a software development kit (SDK) for extensions.

The algorithm proposed by Peterson [1980] was then modified to take into account the fact that our module ranks elements in an already existing list (as provided by OO Writer's spelling checker), as opposed to building this list from scratch. As such, our analysis was restricted only to the elements in that list, instead of looking up some dictionary for spelling alternatives. Upon analysing the words in the list, only words whose edit distance from the wrong word was 1 were considered. The cost of each operation necessary to turn one word in the suggestion list to the wrong word in the text was taken as the probability of finding errors involving that operation in Brazilian Portuguese (cf. Wagner and Fischer [1974]).

This probability, in turn, was estimated based on error frequency in our corpus. A weight is then given to each spelling suggestion, depending on the correction that must be done, according to the frequency of that error in the corpus. Hence, if one typed the word `Nivel` in OO Writer, for example, the suggestion `Nível`[8] would receive weight

[7] "Opinion" in Portuguese.

[8] "Level" in Portuguese.

0.3766, since that is the frequency with which omissions of diacritics happen (see section 4 for a complete list of errors and frequencies). On the non-diacritic side, words such as `traze`[9], for example, would get 0.1712 because that is the probability of a regular omission.

With these statistics at hand, we reviewed the wrong words in the corpus, reordering their suggestion lists, with the higher weighted word in the first position. To do so, all possible variations of each word in Writer's original list are tried, following Damerau [1964]. The algorithm then keeps all these variations in a separate list, assigning proper weights according to the changes made. For example, when an omission is done, this means that the error was an insertion and, thus, a weight which reflects the probability of an insertion to occur is assigned to this suggestion. This is done for all types of errors.

Next, the new list is ordered according to the probability of occurrence of the error. This is repeated for each error type. The list is then further sorted by probability, and finally repetitions are removed. Since this new list contains only words at unitary edit distance from the wrong word, it is possible that the original list delivered by OO Writer may be larger than that. Words belonging to the original list, which are not included in the new list, are then appended to the new list, in the order they appear in the original one.

Since this procedure does actually carry out the testing in the same corpus as statistics were collected, some bias might have been inadvertently introduced. To deal with such a threat to validity, we have applied the same procedure to a different corpus (C_2), collected on June 2011 from four blog posts over the Internet[10], describing travel diaries and comments by visitors. This new corpus comprised 26,418 words, spread over 192 posts, written in Brazilian Portuguese. Tests were then repeated with this corpus, just as before, except that, in this case, error frequencies were those gathered in C_1.

4. Results and Discussion

In order to deal with spelling errors involving diacritics, we have added, to the four categories originally proposed by Damerau, a fifth one: the diacritic error category. Also, because *missing spaces* are hard to deal with, they were classified as an independent error. There is also a sixth category, "others", responsible for grouping up errors related to the making up of expressions (2,02%), to proper nouns starting with non-capital letters (2,72%), and errors related to augmentative and diminutive (1,31%).

Among the errors with diacritics, we found five basic subtypes: (1) missing diacritic, (2) addition of diacritic, (3) right diacritic in the wrong character, (4) wrong diacritic in the right character, and (5) error with `ç` (cedilla). The last one was kept in a separate category because its classification depends on the keyboard layout, as it will be seen next. Table 1 presents these results.

As it can be seen, diacritic-related errors are very frequent (47,14%, cedilla included). Therefore, they cannot be ignored when dealing with texts in Brazilian Portuguese. As such, since missing diacritics, for example, are much more frequent (37.66%) than plain

[9] A misspelling of "trazer" – "to bring" in Portuguese.

[10] `http://bragatte.wordpress.com/`
`http://guilhermebragatte.blogspot.com.br/`
`http://forasteironairlanda.wordpress.com/tag/nomadismo/`
`http://sussuemdublin.wordpress.com/2011/01/`.

Table 1. Frequency and error distribution in the corpus.

Error Category	Errors	Frequency
Insertion	119	10.45%
Omission	195	17.12%
Transposition	42	3.69%
Substitution	146	12.82%
Missing Diacritic	429	37.66%
Addition of Diacritic	19	1.67%
Wrong Diacritic in Right Letter	11	0.96%
Right Diacritic in Wrong Letter	9	0.79%
Missing Cedilla	69	6.06%
Substitution by Space	1	0.09%
Space Insertion	1	0.09%
Space Transposition	2	0.17%
Missing Space	27	2.37%
Other	69	6.06%
Total	1,139	100%

letter substitutions (12.82%), candidates whose turning into the wrong word requires adding a diacritic should be ranked in a higher position than those requiring the substitution of a plain letter. Moreover, should diacritic-related errors be accounted for as substitutions, these would inflate its frequency to almost 60% of the total errors, severely biasing the results.

If, however, we add both Damerau-type errors (44.08%) and diacritic-related errors (47.14%), we end up with a total of 91.22% – not too far from the statistics for English, in which over 80% of the errors are classified into one of Damerau's types. Furthermore, if we let aside errors involving diacritics, we find that 83.39% are acquainted by one of Damerau's categories, confirming his results, as illustrated in Table 2.

Table 2. Frequency and error distribution disregarding diacritics.

Error Category	Errors	Frequency	
Insertion	119	19,77%	83.39%
Omission	195	32.39%	
Transposition	42	6.98%	
Substitution	146	24.25%	
Substitution by Space	1	0.17%	
Space Insertion	1	0.17%	
Space Transposition	2	0.33%	
Missing Space	27	4.48%	
Other	69	11.46%	
Total	602	100%	

A further factor that should be considered when analysing Brazilian Portuguese texts is that at least two types of keyboards are in common use in Brazil: US-Accents e ABNT-2. Depending on the keyboard used, errors with ç could be interpreted in different ways

(either as a diacritic or a substitution error), since ABNT-2 provides a ç key, whereas the US-Accent keyboard does not. In a US-Accent keyboard a ç is a composite, so a change from ç to c may be construed as a diacritic error. On the other hand, if an ABNT-2 keyboard is used, the same change should be considered as a substitution error. Unfortunately, the corpus has no record of the keyboards that were used.

Nevertheless, by moving errors with ç out of the diacritic-related set, one ends up with 41.08% of all errors involving diacritics, instead of 47.14% – still quite a proportion. Also, as far as word correction is concerned, although ç may be considered a substitution error, it may not be interesting to classify it so. For example, when one sees `Cabeca`, it seems more natural that `Cabeça`[11] appears in a better position than `Careca`[12] in the suggestion list. The substitution of `r` for `b` is rather unlikely from an ergonomic viewpoint. Of course, an analysis of the context, along with a more refined model, might further ground this decision.

4.1. Comparing OO Writer's original and new lists

For each wrong word in C_1, we have compared the list generated by our module with OO Writer's original list. Results show that, disregarding word repetitions, *i.e.* accounting for all words, be them repetitions or not (+*Rep* in Table 3), for a total of 1,046 wrongly typed words, our module performed better than OO Writer (*i.e.* it ranked the correct word higher than OO did) in 27.34% of them, whereas in 5.84% it was actually worse (with 66.82% of the words keeping their place in the ranking). On the other hand, by ruling out repetitions (−*Rep* in the Table), from the 509 different words in the corpus, our module performed better in 21.21% of the cases, worse in 11.78%, and as well as OO Writer in 67.01%.

The reason for this difference might rest in the high frequency of words with the same error, as in "nao"[13], for example, which, when reduced to a single erroneous word, reduces the overall accuracy of the system. Whether one should take such repetitions into account is a matter of discussion, and it really depends on the intended use of the statistics. When taking all mistakes into account, one risks biasing the results over words that are constantly mistyped. On the other hand, by ignoring this fact, and taking only different wrong words for the statistics, one looses this information, which might be of interest, depending on the application.

As for C_2, disregarding word repetitions, for a total of 1,055 wrongly typed words, our module performed better than OO Writer in 19.90% of them, whereas in 9.00% it was actually worse (with 71.10% of the words keeping their place in the ranking). By ruling out repetitions, from the 594 different words in the corpus, our module performed better in 20.70% of the cases, worse in 14.47%, and as well as OO Writer in 64.83%. As expected, the performance of our module was reduced when tested in a different corpus. Still, the overall gain (*i.e. OO Worse − OO Better*) was 21.50% for C_1 and 10.90% for C_2, when all the words are accounted for in the testing, and 9.43% for C_1 and 6.23% for C_2, when only different words are included.

Finally, Table 3 also shows that results were close to each other, in both corpora,

[11] "Head" in Portuguese.

[12] "Bald" in Portuguese.

[13] "No" in Portuguese, whose right form would be "não".

Table 3. Comparison between OO Writer's and our new module's results.

	$\mathbf{C}_1$		$\mathbf{C}_2$	
	+Rep.	**−Rep.**	**+Rep.**	**−Rep.**
OO Better	5.84%	11.78%	9.00%	14.47%
OO Worse	27.34%	21.21%	19,90%	20.70%
Ties	66.82%	67.01%	71.10%	64.83%

to the extent that one cannot tell them apart, for the $+$*Rep* and $-$*Rep* categories ($\chi^2 = 0.0449, p = 0.8323$, when word repetitions are ruled out, and $\chi^2 = 0.9015, p = 0.3424$ when they are accounted for). The fact that the discrepancy between both corpora, in the values for $+$*Rep*, is higher than that for $-$*Rep* is but an indication that some wrong words may have been more common in one corpus than in the other.

5. Conclusion

In this report we have shown that, with the aid of statistics tailored to Brazilian Portuguese, in special those related to the treatment of errors involving diacritics, it was possible to improve over the ranking of suggestions for spelling correction in a commercially available text editor. Through a straightforward approach, which gives weight to suggestions according to the frequency with which errors take place in that language, it was possible to observe an improvement that ranges from around 6.2% to 21.5% in the order spelling suggestions are presented to the user, depending on the testing corpus and whether or not one takes only different wrong words into account (as opposed to all mistakes made in the corpus).

Our intention with this experiment was to provide an insight on the practical usefulness of these statistics alone. Naturally, more elaborated methods, such as taking into account suggestions farther than unitary edit distance away from the wrong word, for example, or even relying on the context for breaking ties, might lead to better results. Still, such an improvement for so low a price certainly deserves some attention.

References

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176.

Deorowicz, S. and Ciura, M. G. (2005). Correcting spelling errors by modeling their causes. *International Journal of Applied Mathematics and Computer Science*, 15(2):275–285.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, New Jersey.

Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439.

Medeiros, J. C. D. (1995). Processamento morfológico e correc cão ortográfica do português. Master's thesis, Instituto Superior Técnico – Universidade Técnica De Lisboa.

Peterson, J. L. (1980). Computer programs for detecting and correcting spelling errors. *Communications of the ACM*, 23(12):676–687.

Peterson, J. L. (1986). A note on undetected typing errors. *Communications of the ACM*, 29(7):633–637.

Roman, N. T., Piwek, P., Carvalho, A. M. B. R., and Alvares, A. R. (2013). Introducing a corpus of human-authored dialogue summaries in portuguese. In *Proceedings of the 2013 International Conference Recent Advances in Natural Language Processing (RANLP 2013)*, pages 692–701, Hissar, Bulgaria.

van Berkel, B. and Smedt, K. D. (1988). Triphone analysis: a combined method for the correction of orthographical and typographical errors. In *Proceedings of the 30th annual meeting of the association for computational linguistics (ACL)*, pages 77–83, Austin, Texas, USA.

Veronis, J. (1988). Computerized correction of phonographic errors. *Computers and the Humanities*, 22(1):43–56.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173.