A RULE-BASED SYLLABIFICATION ALGORITHM WITH STRESS DETERMINATION FOR BRAZILIAN PORTUGUESE NATURAL LANGUAGE PROCESSING

Anderson Monte[a], *Danielle Ribeiro*[b], *Nelson Neto*[a], *Regina Cruz*[b] *& Aldebaro Klautau*[a]

[a]Signal Processing Laboratory (LaPS), Federal University of Pará (UFPA), Brazil;
[b]Institute of Letters and Communication (ILC), Federal University of Pará (UFPA), Brazil
aomonte@ufpa.br; nelsonneto@ufpa.br; regina@ufpa.br;
aldebaro@ufpa.br; dany.priscyla@gmail.com

ABSTRACT

This paper presents some improvements on an existing set of linguistic rules that is capable of performing the syllabification of Brazilian Portuguese words. An algorithm was also implemented and based on this set, which improvements previously mentioned include new rules that depend on the stressed vowel to achieve the standard syllabification of some words that otherwise would be very difficult to do so, when just the graphemes themselves are considered for the process mentioned above.

With the improvements applied to the original rule set, the new version of the algorithm achieved a better performance than the previous one (without the improvements) in performing the syllabification of four different kinds of diphthongs that exist in Brazilian Portuguese.

Keywords: syllabification, stressed vowel, grapheme, Brazilian Portuguese, text-to-speech

1. INTRODUCTION

Text-to-speech (TTS) systems are considered not just very innovative, but also a very mature technology in the speech area [3]. It consists in converting natural language texts into synthesized speech. To make these kinds of systems robust, efficient, and reliable, it is crucial to have a good pre-processing module for the word corpus to be used in the training of a TTS system, as well as for the texts to be synthesized.

The pre-processing module (also known as front-end) of a TTS system is composed by three stages: text analysis, phonetic analysis and prosodic generation [2]. The syllabification of a word is a task for which the text analysis stage is responsible, therefore the algorithm referenced by this paper is intended to help improving the overall efficiency of Brazilian Portuguese (BP) TTS systems and of any other kind of intelligent system that requires analysis of texts, written in the previously mentioned language.

This paper is organized as follows: Section 2 provides an explanation regarding the proposed syllabification algorithm and how it was implemented. Section 3 describes the new rules and the existing ones that were updated to improve the syllabification task. Section 4 presents the results of tests, comparing the efficiency of the old and the new version of the algorithm, and the syllabification made by a web dictionary [4], as a reference. Finally, in Section 5, the conclusion is presented.

2. THE ALGORITHM AND ITS IMPLEMENTATION

The first syllabification algorithm, implemented in this work, was based in the original set of 20 linguistic rules that can be found in [6]. In addition to this, such algorithm and its new version were entirely implemented in Java. They are open-source and available free of cost at [5] for any person or research group that wants to use them.

The rule-based approach was chosen for this work because it has, according to [6], two advantages when compared to the dictionary-based approach: it accepts new words added to a vocabulary and it does not need to keep a large file containing all the words of a certain language (the use of such file eventually requires a large memory).

2.1. Auxiliary methods

It is important to state that the original linguistic rules verify not only the arrangement, but also the kind of some graphemes in certain positions. Because of the second verification, a number of auxiliary methods was implemented to help each rule verify if a certain grapheme, in a given position, belongs to one of the kinds required by that rule. Each one of these methods verify if a

certain grapheme, at a given position (passed as an argument), belongs to one of the following kinds: vowel, semi-vowel or consonant. There is also another auxiliary methods that verify, when it is required by a rule, if a certain grapheme belongs to one of the following specific kinds of consonant: occlusive, fricative, liquid or nasal.

2.2. Syllabification actions

All the rules were implemented in the algorithm to be inside a loop that executes as many iterations as possible to have all the syllables of a given word separated, always analyzing such word from left to right. On each iteration, the algorithm attempts to find a new syllable in the portion of the word that has not yet been analyzed by trying to match one of the rules with this entire portion or just part of it, always obeying the hierarchical sequence for the verification of these rules defined in [6]. When a rule is matched, depending on its definition, the algorithm can take one of the six actions described in [6] to extract a new syllable from that part of the word that is currently under analysis.

These syllabification actions generally work as follows: the method that implements an action receives (as an argument) the position of the first occurrence of a vowel in the non-analyzed portion of a given word (it is necessary because, according to [1], the syllabification process for BP words must consider that every syllable of a given word must have a vowel as a nucleus, and this vowel can be surrounded by consonants, glides or other vowels).

As soon as the argument is passed, the method starts its execution by creating a "String" Java object that is immediately used in two independent loops that check each grapheme in the given word until it reaches a certain position that depends on the position passed as an argument (they can be the same position or differ by one or more graphemes). The goal of such loops is to detect the graphemes that does not have a syllable yet and compose a new syllable with these graphemes, saving the newly composed syllable in the "String" object.

After the conclusion of the loops, the method updates some variables that are responsible for the internal control of the algorithm, including one that saves the new position where the non-analyzed portion of the given word must start, after the last syllable division done (it is important to remember that it becomes smaller as more syllables are separated). Finally, the method finishes its execution by returning the "String" object containing the new syllable found.

2.3. Implementation of the linguistic rules

Each linguistic rule of the algorithm is basically composed of a condition to be evaluated and one of the six syllabification actions. The condition evaluates, in the non-analyzed part of a given word, the kind and the arrangement of graphemes that surrounds the nucleus vowel currently under analysis. If it is fulfilled, then the algorithm calls the method that executes the action associated to such rule to perform the required syllabification. Some rules are composed of more than one action, but they also have specific conditions to help the algorithm decide which action must be taken, once the main condition is fulfilled.

3. NEW LINGUISTIC RULES AND IMPROVEMENTS

In order to improve the efficiency of the syllabification process, we proposed and implemented two new rules for a new version of the algorithm described in Section 2. In addition to that, the rule 19 was updated to fix some errors that occurred when it was previously proposed in [6].

3.1. The new rules

The original syllabification rules, as previously mentioned, consider the kind and the arrangement of graphemes to separate the syllables of a given word. However, there are some words whose syllabification is very difficult to perform correctly with only these two criteria, especially when such words have diphthongs, because they will require a number of very specific and well elaborated rules, in which each one will deal with just a few examples.

To overcome this difficulty, two new linguistic rules, shown below, were proposed, each one not just considering the graphemes themselves, but also their stress. The first rule was proposed for the falling diphthongs (the "vowel + glide" combination), while the second one deals with diphthongs that varies with hiatus (the "glide + vowel" combination). Due to the fact that diphthongs need this special treatment in their syllabification, we defined that these rules must be evaluated before the 20 original ones.

The main motivation for analyzing the previously mentioned diphthongs comes from the perception of existing divergences between the

scholars on such subject (like the position of a glide, inside a syllable, in the falling diphthongs, that was explained in [1]). Another point that ratifies the focus adopted in this analysis is the fact that vocalic segments, especially the ones with rising sonority, have presented more errors in the separations performed by the syllabification algorithm (in our previous analysis). It is important to highlight that such imprecisions were detected taking into account the pauses produced in the speech of a person from Belém (the capital of Pará, Brazil), since the dialectal difference influences the fact whether a word is a diphthong or a hiatus.

Falling diphthongs rule: (<a>, <e>, <o>) + (<i>, <u>):

- **If (<i> or <u>) is the stressed grapheme:**
  - **Then, (<a>, <e> or <o>) must be separated from (<i> or <u>).** Examples: sa-í-da, gra-ú-do.
  - **Else, (<a>, <e> or <o>) and (<i> or <u>) must stay in the same syllable.** Examples: cãi-bra, mai-se-na.

Diphthongs that varies with hiatus rule: (<i>, <u>) + (<a>, <e>, <o>):

- **If such combination is not at the end of the word:**
  - **Then, (<i> or <u>) must be separated from (<a>, <e> or <o>).** Example: bi-o-ma.
  - **Else, the following condition must be evaluated: if (<i> or <u>) or (<a>, <e> or <o>) is the stressed grapheme:**
    - **Then, (<i> or <u>) must be separated from (<a>, <e> or <o>).** Examples: de-mo-cra-ci-a, ta-man-du-á
    - **Else, (<i> or <u>) and (<a>, <e> or <o>) must stay in the same syllable.** Examples: só-cio, cí-lio.

3.2. The updated rule 19

The updated version of the rule 19 is shown below.

Rule 19:

- **If the analyzed vowel is not the first grapheme in the next syllable to be formed and is followed by another vowel that precedes a consonant (vowel + vowel + consonant combination):**
  - **Then, the analyzed vowel must be separated from the following graphemes**.

This new version of the rule 19 fixes some errors that were occurring in the syllabification of words like "teólogo", for example (the correct is "te-ó-lo-go", instead of "teó-lo-go", as shown in [6]).

4. TESTS AND RESULTS

This work used 170 words in total that, aiming to delimitate the encountered problems, were divided in four contexts: falling diphthongs, rising diphthongs (compounds 'kwa', 'kwe', 'kwo' or 'gwa', 'gwe'), diphthongs that varies with hiatus and false diphthongs. For each word, three syllabifications were analyzed and extracted from the following sources: the web dictionary, the old version of the syllabification algorithm (without the new rules) and the new version of the same algorithm, with the new rules implemented. The Table 1 shows the results of the syllabification tests for each source (the divergences between the syllabifications performed by the sources and the standards for syllable separation in BP, described in [1], were considered as errors encountered during the syllabification process of these sources).

Table 1: Error percentage analysis of the syllabification sources.

Source	Words in Total	Errors Occurred	Error Percentage
Web Dictionary	170	1	0,58%
Syllabification Algorithm (Old)	170	29	11,17%
Syllabification Algorithm (New)	170	2	1,17%

According to the results shown in Table 1, the old version of the syllabification algorithm was the source with the most occurrence of errors during the tests. In the context of the falling diphthongs, four words have presented problems: "cãibra", "cílio", "eutanásia" "and" "saída". In the former three ones, the mistake is in the fact that such words are composed of vocalic encounters formed by a vowel and a glide, and the syllabifications "cã-i-bra", "cí-li-o" and "eu-ta-ná-si-a" create syllable nucleus composed by a glide. According to [1], this can not be possible because, in the BP words, the nucleus of a syllable must always be a vowel, while the glides must appear in syllable coda, in falling diphthongs. Due to such condition, the accepted syllabifications are: "cãi-bra", "cí-lio" and "eu-ta-ná-sia".

The syllable division "saí-da" is incorrect, because the word has an hiatus, and not a diphthong, since it presents two vowels, where the first one is low (/a/) and the second one is high

(/i/). The use of graphic accent makes this word an hiatus for excellence, therefore the accepted syllabification is “sa-í-da”.

In the context of rising diphthongs formed by ‘kwa’, ‘kwe’, ‘kwo’ or ‘gwa’, ‘gwe’, two words have presented problems: “guelra” and “quotidiano”. In the syllable division “gue-lra”, the error is not at the vocalic incidence, it is located at consonantal encounter, once the Portuguese grammar does not accept the /lr/ occupying, simultaneously, the onset of a syllable, what can be justified by the sonority sequence, responsible for the correlation of each syllabic segment present in a word.

The syllabification “quo-ti-dia-no” is wrong because of the second vocalic encounter (/ia/), that has two vowels. Actually, they must belong to different syllables to be capable of composing nuclei.

Regarding the diphthongs that varies with hiatus, 23 words presented errors. The syllable divisions “a-lua-men-to”, “ba-bu-í-no”, “ciú-me”, “cue-ca”, “fia-dor”, “mia-do”, “mio-lo”, “na-cio-nal”, “pia-da”, “pie-da-de”, “pie-gas”, “pie-tis-ta”, “pio-la”, “sua-do”, “sua-ve”, “suí-de-o”, “suí-no”, “ria-cho”, “ta-tua-gem”, “vi-ú-va” and “zoa-da” are incorrect because these words has hiatus, since they present two vowels, where the first one is high (/i/ or /u/) and the second one is low or mid. It is important to emphasize that, like “saída”, the words “babuíno”, “ciúme”, “suídeo”, “suíno” and “viúva” composes the group that leaves no doubts about being hiatus because of the graphic accent in the written form of these words. The syllable division “juí-zo” is also incorrect because it has an hiatus formed by two high vowels (/i/ and /u/) instead of a diphthong.

The syllabification “his-tó-ri-a” is wrong because it has a diphthong, since the vocalic encounter is composed by a glide and a vowel, and not two vowels. The accepted division is “his-tó-ria”.

Regarding the false diphthongs, there has not occurred any errors.

The new version of the syllabification algorithm achieved a very low amount of errors, compared to its previous version. The mentioned errors were detected in two words in the context of the falling diphthongs: “eufonia” and “ousadia”.

The syllable divisions “eu-fo-nia” and “ou-sa-dia” are incorrect because each one presents two vowels that should be separated to compose nuclei for two different syllables. The accepted syllabifications are: “eu-fo-ni-a” and “ou-sa-di-a”.

Regarding the test made with the web dictionary, only one error was encountered and it occurred in the word “maisena”. Like “cãibra”, “cílio” and “eutanásia”, the word “maisena” presents a composition formed by a vowel and a glide, what does not allow the separation of the vocalic encounter because, as said before, it is not possible, in BP, to make a nucleus from a glide. The accepted syllable division is “mai-se-na”.

5. CONCLUSION

In this work, we presented new rules for improving the original rule set proposed in [6]. Although the goal of two of these new rules is to perform the correct syllabification of BP words with falling diphthongs and diphthongs that varies with hiatus, the new version of the rule set (that includes such new rules) has proven to be also very efficient with words containing rising diphthongs and false diphthongs, achieving an error percentage of only 1,17% in a test that used 170 words involving all the four kinds of diphthongs mentioned here.

As future work, we intend to adjust the existing rules or create new ones, making use of the information about the stressed vowel of a word, if possible, and realize new tests using tens of thousands of words besides the ones containing diphthongs and hiatus, to evaluate the efficiency of the algorithm against words of all kinds.

6. REFERENCES

[1] Bisol, L. 2005. *Introdução a Estudos de Fonologia do Português Brasileiro*. Porto Alegre: EDIPUCRS.

[2] Braga, D., Dias, M.S. 2009. Sistemas de Conversão Texto-Fala: Estado da arte, aplicações, arquitectura e desafios. *http://pt.scribd.com/doc/32579516/Braga- Texto-Fala*

[3] Couto, I., Neto, N., Tadaiesky, V., Klautau, A., Maia, R. 2010. An open source HMM-based text-to-speech system for Brazilian Portuguese. *Proc. 7th International Telecommunications Symposium* Manaus.

[4] Dicionário Web. *http://www.dicionarioweb.com.br/*

[5] Projeto FalaBrasil. *http://www.laps.ufpa.br/falabrasil/*

[6] Silva, D.C., Braga, D., Resende, F.G.V. 2008. Separação das Sílabas e Determinação da Tonicidade no Português Brasileiro. *Proc. XXVI Simpósio Brasileiro de Telecomunicações* Rio de Janeiro, 1-5.